

BEST AVAILABLE COPY

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-182012

(43)Date of publication of application : 30.06.2000

(51)Int.Cl.

G06K 19/073

G09C 1/00

H04L 9/10

(21)Application number : 10-354156

(71)Applicant :

HITACHI LTD

(22)Date of filing : 14.12.1998

(72)Inventor :

OKI MASARU

FUKUZAWA YASUKO

OKUHARA SUSUMU

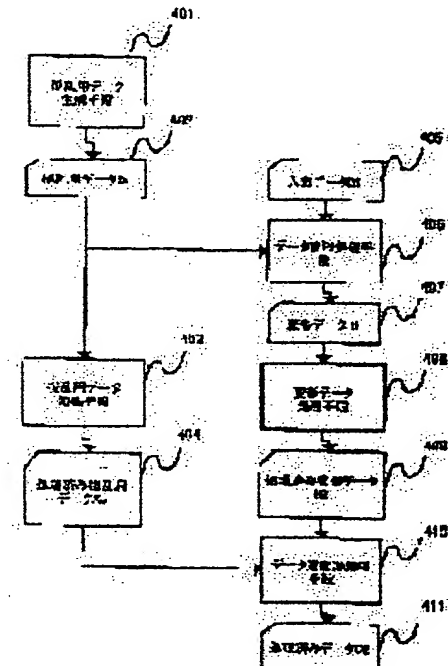
KAMINAGA MASAHIRO

(54) INFORMATION PROCESSOR AND END TAMPER PROCESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To make it possible to interence processing and a ciphering key from the waveform of current consumption by deforming processing data with disturbing data, processing data with deformed data and inversely deforming it through the use of disturbing data after processing to obtain a correct processing result in an IC card chip so as to reduce relation between data processing and current consumption in the IC card chip.

SOLUTION: A disturbing data generating means 401 generates disturbing data Xi. Next, a data deforming processing means 406 deforms an inputted data D1 (405) with disturbing data Xi to generate deforming data H1 (407). Then, through the use of data H1, a deformed data processing means 408 executes data processing to generate processed deforming data H2 (409). On the other hand, the data Xi is given data processing equal to inputted data by a disturbing data processing means 403 to prepare processed disturbing data Xo (404). True processed data D2 (411) is obtained by a data inversely deform processing means 410 through the use of this data Xo and data H2.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japanese Patent Office

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開2000-182012

(P2000-182012A)

(43)公開日 平成12年6月30日(2000.6.30)

(51)Int.Cl. ⁷	識別記号	F I	テマコード [*] (参考)
G 0 6 K 19/073		G 0 6 K 19/00	P 5 B 0 3 5
G 0 9 C 1/00	6 6 0	G 0 9 C 1/00	6 6 0 A 5 J 1 0 4
H 0 4 L 9/10		H 0 4 L 9/00	6 2 1 Z 9 A 0 0 1

審査請求 未請求 請求項の数23 O L (全 26 頁)

(21)出願番号 特願平10-354156

(22)出願日 平成10年12月14日(1998.12.14)

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 大木 優

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72)発明者 福澤 幸子

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム研究所内

(74)代理人 100068504

弁理士 小川 勝男

最終頁に続く

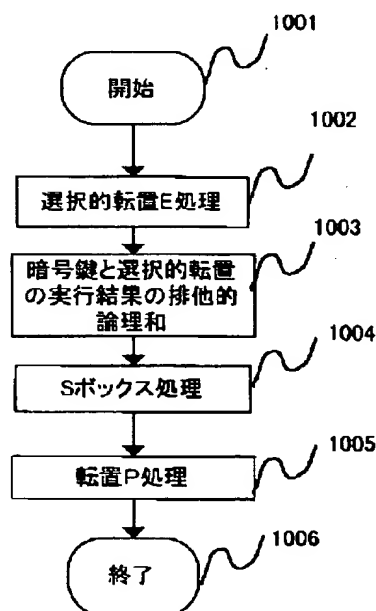
(54)【発明の名称】 情報処理装置、端タンバ処理装置

(57)【要約】

【課題】 ICカード用チップのプログラムでの処理を行う前に変形を加え、処理が終わった後に逆変形を行い、正しい処理結果を得る。ICカード用チップでの処理内容とICカード用チップの消費電流との関連性を減らす。

【解決手段】 ICカード用チップの処理を行う前にを行う前に変形を加えることにより、処理データとICカード用チップの消費電流との関連性を減らす。処理が終わった後に逆変形を行い、正しい処理結果を得る。ICカード用チップでの処理のデータとICカード用チップの消費電流との関連性を減らすことを可能とする。

図10



【特許請求の範囲】

【請求項1】プログラムを格納するプログラム格納部、データを保存するデータ格納部を持つ記憶装置とプログラムに従い、所定の処理を実行し、データ処理を行う中央演算装置を持ち、プログラムは、中央演算装置に実行の指示を与える処理命令から構成される一つ以上のデータ処理手段からなり、一つのデータ処理手段が、その入力データを処理し、処理済みデータを出力する入力データ処理手段を含む、情報処理装置において、攪乱用データXiを使って入力データD1を変形し、変形データH1を作成するデータ変形処理手段、変形データH1を入力データ処理手段と同じ処理を行い、処理済み変形データH2を作成する変形データ処理手段、攪乱用データXiを入力データ処理手段と同じ処理を行い、処理済み攪乱用データXoを作成する攪乱用データ処理手段、処理済み攪乱用データXoを使って処理済み変形データH2を処理し、入力データD1を入力データ処理手段で処理した結果である処理済みデータD2を得るデータ逆変形処理手段、を有することを特徴とする情報処理装置。

【請求項2】プログラムを格納するプログラム格納部、データを保存するデータ格納部を持つ記憶装置とプログラムに従い、所定の処理を実行し、データ処理を行う中央演算装置を持ち、プログラムは、中央演算装置に実行の指示を与える処理命令から構成される一つ以上のデータ処理手段からなり、一つのデータ処理手段が、その第1入力データを処理し、第1処理済みデータを出力する第1入力データ処理手段と、別のデータ処理手段が、その第2入力データを処理し、第2処理済みデータを出力する第2入力データ処理手段を含む、情報処理装置において、第1攪乱用データX1iを使って第1入力データD1を変形し、変形データH1を作成する第1データ変形処理手段、変形データH1を第1入力データ処理手段と同じ処理を行い、処理済み変形データH2を作成する第1変形データ処理手段、第1攪乱用データX1iを第1入力データ処理手段と同じ処理を行い、第1処理済み攪乱用データX1oを作成する第1攪乱用データ処理手段、第2攪乱用データX2iを使って処理済み変形データH2を変形し、処理済み変形データH3を作成する第2データ変形処理手段、処理済み変形データH3を第2入力データ処理手段と同じ処理を行い、処理済み変形データH4を作成する第2変形データ処理手段、第2攪乱用データX2iを第2入力データ処理手段と同じ処理を行い、第2処理済み攪乱用データX2oを作成する第2攪乱用データ処理手段、第2処理済み攪乱用データX2oを使って処理済み変形データH4を処理し、処理済み変形データH5を得る第2データ逆変形処理手段、第1処理済み攪乱用データX1oを使って処理済み変形データH5を処理し、処理済みデータD2を得る第1データ逆変形処理手段、を有することを特徴とする情報処理装置。

【請求項3】プログラムを格納するプログラム格納部、

データを保存するデータ格納部を持つ記憶装置とプログラムに従い、所定の処理を実行し、データ処理を行う中央演算装置を持ち、プログラムは、中央演算装置に実行の指示を与える処理命令から構成される一つ以上のデータ処理手段からなり、一つのデータ処理手段が、その第1入力データを処理し、第1処理済みデータを出力する第1入力データ処理手段と、別のデータ処理手段が、その第2入力データを処理し、第2処理済みデータを出力する第2入力データ処理手段を含む、情報処理装置において、第1攪乱用データX1iを使って第1入力データD1を変形し、変形データH1を作成する第1データ変形処理手段、変形データH1を第1入力データ処理手段と同じ処理を行い、処理済み変形データH2を作成する第1変形データ処理手段、第1攪乱用データX1iを第1入力データ処理手段と同じ処理を行い、第1処理済み攪乱用データX1oを作成する第1攪乱用データ処理手段、第2攪乱用データX2iを使って処理済み変形データH2を変形し、処理済み変形データH3を作成する第2データ変形処理手段、第1処理済み攪乱用データX1oを使って処理済み変形データH3を処理し、処理済み変形データH4を得る第1データ逆変形処理手段、処理済み変形データH4を第2入力データ処理手段と同じ処理を行い、処理済み変形データH5を作成する第2変形データ処理手段、第2攪乱用データX2iを第2入力データ処理手段と同じ処理を行い、第2処理済み攪乱用データX2oを作成する第2攪乱用データ処理手段、第2処理済み攪乱用データX2oを使って処理済み変形データH5を処理し、処理済みデータD2を得る第2データ逆変形処理手段、を有することを特徴とする情報処理装置。

【請求項4】プログラムを格納するプログラム格納部、データを保存するデータ格納部を持つ記憶装置とプログラムに従い、所定の処理を実行し、データ処理を行う中央演算装置を持ち、プログラムは、中央演算装置に実行の指示を与える処理命令から構成される一つ以上のデータ処理手段からなり、一つのデータ処理手段が、その第1入力データを処理し、第1処理済みデータを出力する第1入力データ処理手段と、別のデータ処理手段が、その第2入力データを処理し、第2処理済みデータを出力する第2入力データ処理手段を含む、情報処理装置において、第1攪乱用データX1iを使って第1入力データD1を変形し、変形データH1を作成する第1データ変形処理手段、変形データH1を第1入力データ処理手段と同じ処理を行い、処理済み変形データH2を作成する第1変形データ処理手段、第1攪乱用データX1iを第1入力データ処理手段と同じ処理を行い、第1処理済み攪乱用データX1oを作成する第1攪乱用データ処理手段、第2攪乱用データX2iを使って処理済み変形データH2を変形し、処理済み変形データH3を作成する第2データ変形処理手段、処理済み変形データH3を第2入力データ処理手段と同じ処理を行い、処理済み変形データH4を作成する第2

変形データ処理手段、第2撹乱用データX2iを第2入力データ処理手段と同じ処理を行い、第2処理済み撹乱用データX2oを作成する第2撹乱用データ処理手段、第1処理済み撹乱用データX1oと第2処理済み撹乱用データX2oの逆変形を統合し、統合処理済み撹乱用データXoを作成するデータ逆変形統合処理手段、統合処理済み撹乱用データXoを使って処理済み変形データH4を処理し、処理済みデータD2を得るデータ逆変形処理手段、を有することを特徴とする情報処理装置。

【請求項5】請求項1、2、3、4において、撹乱用データXiを入力データ処理手段と同じ処理を行い、処理済み撹乱用データXoを作成する撹乱用データ処理手段、処理済み撹乱用データXoを記録する処理済み撹乱用データ記録手段、記録した処理済み撹乱用データXoを使って処理済み変形データを処理し、新たに別の処理済み変形データを得るデータ逆変形処理手段、を有することを特徴とする情報処理装置。

【請求項6】請求項1、2、3、4、5において、撹乱用データとして乱数を生成する隠蔽データ生成処理手段、を有することを特徴とする情報処理装置。

【請求項7】請求項1、2、3、4、5、6において、データ変形処理手段およびデータ逆変形処理手段として、排他的論理和を使うことを特徴とする情報処理装置。

【請求項8】請求項1、2、3、4、5、6において、データ変形処理手段として、加算を使うことデータ逆変形処理手段として、減算を使うことを特徴とする情報処理装置。

【請求項9】請求項1、2、3、4、5、6において、データ変形処理手段として、減算を使うことデータ逆変形処理手段として、加算を使うことを特徴とする情報処理装置。

【請求項10】請求項1、2、3、4、5、6において、データ変形処理手段として、掛け算を使うことデータ逆変形処理手段として、割り算を使うことを特徴とする情報処理装置。

【請求項11】請求項1、2、3、4、5、6において、データ変形処理手段として、割り算を使うことデータ逆変形処理手段として、掛け算を使うことを特徴とする情報処理装置。

【請求項12】請求項1、2、3、4、5、6において、データ変形処理手段として、剰余演算で剰余演算の法Nの整数倍を加算することデータ逆変形処理手段が必要であることを特徴とする情報処理装置。

【請求項13】請求項1、2、3、4、5、6において、剰余演算で

$$I = X * Y \text{ mod } N$$
という関係のある数X、Yにおいてデータ変形処理手段として、剰余演算でXを整数倍乗算することデータ逆変形処理手段として、データ変形処理手段でXを乗算した回

数分、Yを整数倍乗算することを特徴とする情報処理装置。

【請求項14】請求項1、2、3、4、5、6において、データ変形処理手段として、Nを剰余演算の法として、(N+1)/2を整数倍乗算すること、データ逆変形処理手段として、データ変形処理手段で(N+1)/2を乗算した回数分、2を整数倍乗算することを特徴とする情報処理装置。

【請求項15】請求項1、2、3、4、5、6において、データ変形処理手段として、剰余演算で2を整数倍乗算することデータ逆変形処理手段として、データ変形処理手段で2を乗算した回数分、Nを剰余演算の法として、(N+1)/2を整数倍乗算することを特徴とする情報処理装置。

【請求項16】請求項1、2、3、4、5、6において、データ変形処理手段として、配列のデータの配置を規則的な方法で変更すること、データ逆変形処理手段として、配列のデータの配置が変更された方法で配列のデータをアクセスすることを特徴とする情報処理装置。

【請求項17】請求項16において、データ変形処理手段での配列のデータの配置を規則的に変更方法として、配列のインデックス（引数）のある数で排他的論理和をとり、データの配置を入れ替えること、データ逆変形処理手段で配列のデータにアクセスする際、配列のインデックスをデータ変形処理手段で使用した数で排他的論理和をとり、変形後の配列のインデックスを作成し、アクセスすることを特徴とする情報処理装置。

【請求項18】請求項1から6において、データ処理手段がビット単位のデータを入れ替える転置処理手段であること、データ変形処理手段及びデータ逆変形処理手段が排他的論理和であることを特徴とする情報処理装置。

【請求項19】請求項1から6において、データ処理手段がバイト単位のデータを入れ替える換字処理手段であること、データ変形処理手段及びデータ逆変形処理手段が排他的論理和であることを特徴とする情報処理装置。

【請求項20】請求項1から6において、データ処理手段がテーブルを使ってデータを入れ替える処理手段であること、データ変形処理手段及びデータ逆変形処理手段が排他的論理和であることを特徴とする情報処理装置。

【請求項21】請求項3において、第1のデータ処理が、データ1とデータ2の排他的論理和を行いデータ3を生成する、排他的論理和処理手段であること、第2のデータ処理が、データ3を用いて、配列のインデックスを計算する配列アクセス処理手段であること、第1の変形処理手段として、第1の撹乱用データとデータの排他的論理和を使うこと、第1のデータ逆変形処理手段として、排他的論理和を使うこと、第2の変形処理手段として、第2の撹乱用データと配列のインデックスを排他的論理和をとること、第2のデータ逆変形処理手段として、排他的論理和を使うこと、を特徴とする情報処理装

置。

【請求項22】請求項1、2、3、4、5、6において、データ変換処理手段として、剰余演算で剰余演算の法Nの整数倍に1を加えたものを乗算すること、データ逆変換処理手段が不要であることを特徴とする情報処理装置。

【請求項23】データを保存する記憶装置と、攪乱用データを生成する攪乱用データ生成手段と、上記データを上記攪乱用データを用いて変形して変形データを作成する変形データ作成手段と、上記変形データを処理する変形データ処理手段と、上記攪乱用データを処理する攪乱用データ処理手段と、処理後の上記変形データを処理後の上記攪乱用データを用いて逆変形する逆変形手段とを有することを特徴とする耐タンパ処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、機密性の高いICカードなどの耐タンパ装置に関するものである。

【0002】

【従来の技術】ICカードは、主に、勝手に書き換えられない情報の保持や秘密情報である暗号鍵を使ったデータの暗号化や暗号文の復号化を行うために使われる装置である。ICカードは、電源を持っていないため、リーダライタに差し込まれると、電源の供給を受け、動作可能となる。動作可能になると、リーダライタからコマンドを受けて、コマンドに従い、データの転送を行う。ICカードの一般的な解説は、オーム社出版電子情報通信学会編水沢順一著「ICカード」などにある。

【0003】ICカードの構成は、図1に示すように、カード101の上に、ICカード用チップ102を搭載したものである。一般にICカードは、接点を持ち、接点を通して、リーダライタから電源の供給やリーダライタとのデータの通信を行う。

【0004】ICカード用チップの構成は、基本的にマイクロコンピュータと同じような構成である。その構成は、図2に示すように、中央演算装置201、記憶装置204、入出力ポート207、コ・プロセッサ202からなる。中央処理装置201は、論理演算や算術演算などを行う装置であり、記憶装置204は、プログラムやデータを格納する装置である。入出力ポートは、リーダライタと通信を行う装置である。コ・プロセッサは、剰余演算を行うための特別な演算装置であり、非対称暗号であるRSAの演算などに用いられる装置である。ICカード用プロセッサの中には、コ・プロセッサを持たないものも多くある。データバス203は、各装置を接続するバスである。

【0005】記憶装置204は、ROM(Read Only Memory)やRAM(Random Access Memory)、EEPROM(Electrical Erasable Programmable Read Only Memory)などからなる。ROMは、変更できないメモリであり、主にプログラ

ムを格納するメモリである。RAMは自由に書き換えができるメモリであるが、電源の供給が中断されると、記憶している内容が消えてなくなる。ICカードがリーダライタから抜かれると電源の供給が中断されるため、RAMの内容は、保持できなくなる。EEPROMは、電源の供給が中断されてもその内容を保持することができるメモリである。書き換える必要があり、ICカードがリーダライタから抜かれても、保持するデータを格納するために使われる。例えば、プリペイドカードでのプリペイドの度数などは、使用するたびに書き換えられ、かつリーダライタか抜かれてもデータを保持する必要があるため、EEPROMで保持される。

【0006】ICカードは、プログラムや重要な情報がICカード用チップの中に密閉されているため、重要な情報を格納したり、カードの中で暗号処理を行うために、使われている。ICカードでの暗号処理の解読の難しさは、暗号アルゴリズムの解読の困難さと同じと考えられていた。しかし、ICカードが暗号処理を行っている時の消費電流を観測し、解析することにより、暗号アルゴリズムの解読より容易に暗号処理の内容や暗号鍵が推定される可能性が示唆されている。消費電流は、リーダライタから供給されている電流を測定することにより観測することができる。このような危険性は、John Wiley & sons社 W.Rankl & W.Effing著「Smart Card Handbook」の8.5.1.1 Passive protective mechanisms(263ページ)に記載されている。

【0007】ICカード用チップを構成しているCMOSは、出力状態が1から0あるいは0から1に変わった時に電流を消費する。特に、データバス203のバスは、大きな電気容量を持つため、バスの値が1から0あるいは0から1に変わると、大きな電流を消費する。そのため、消費電流を観測すれば、ICカード用チップの中で、何が動作しているか分かる可能性を示唆している。

【0008】図3は、ICカード用チップの1サイクルでの消費電流の波形を示したものである。処理しているデータの依存して、電流波形が301や302のように異なる。このような差は、バス203を流れるデータや中央演算装置201で処理しているデータに依存して生じる。

【0009】16ビットのプリチャージバスにデータを転送する場合を考える。プリチャージバスは、データ転送の前にすべてのバスの値を“0”にそろえるバスである。このバスに、値は違うが“1”のビットの数が同じデータ、例えば、“1”のビットの数が2である16進数で“88”と“11”、を転送した場合、電流波形はほぼ同じ波形になる。この理由は、“0”から“1”へ変化したビットの数が同じであるため、同じように電流をし消費し、同じ電流波形になったからである。もし、“1”のビットの数が1つ異なるデータ、例えば、“1”のビットの数が3である“89”や“19”を転

送した場合、“1”のビットの数が2のデータとは消費電流が異なる。これは、3ビット分バスの値が“0”から“1”に変わったため、その分の電流が消費される。そのため、先の2ビットが変化したデータに比べて消費電流が1ビット分大きくなる。一般に、“1”のビットの数が多いほど電流波形は高くなるという規則性がある。この規則性から、転送されているデータを推定することができる。

【0010】図3の電流波形は、バスだけではなく、ICカード用チップを構成している構成要素の消費している電流の総和である。しかし、ICカード用チップのようなマイコンでは、主にバスにデータが転送されているフェーズ、主にCPUが演算をしているフェーズ、レジスタにデータを書き込んでいるフェーズなどに分かれている。そのため、フェーズに注目すれば、主にどの部分での消費電流の違いが分かり、その部分でのデータ操作を推定することができる。

【0011】具体的な命令でどのように差が出るかを、次のような左シフト命令を例にして説明する。

【0012】shiftl R1 (式1)

この命令は、レジスタ1の内容を左シフトし、レジスタのビット列が左にシフトし、最上位ビットの値がキャリアとしてコンディションコードレジスタに入る命令である。レジスタR1の最上位ビットがデータバスを経由して、コンディションコードレジスタに転送されるため、電流波形の大きさを比較すれば、最上位のビットが“0”か“1”かが識別できる可能性がある。もしR1に重要なデータが入っていれば、そのデータの1ビットであるが、“0”か“1”かが分かる可能性がある。特に、DESのような暗号処理では、暗号鍵をシフトする操作が頻発する。このシフト操作のときに暗号鍵のデータを推定できる電流波形が生じ、暗号鍵を推定される危険性がある。

【0013】これは、コ・プロセッサ202の演算でも同じである。演算内容が、暗号鍵に依存した偏りがあると、その偏りが消費電流から求められ、暗号鍵が推定される可能性がある。

【0014】

【発明が解決しようとする課題】本発明の課題は、ICカード用チップでのデータ処理と消費電流との関連性を減らすことである。消費電流とチップの処理との関連性が減れば、観測した消費電流の波形からICカードチップ内での処理や暗号鍵の推測が困難になる。本発明の着眼点は、ICカードチップでの処理するデータを変形して処理することにより、消費電流の波形から、処理や暗号鍵の推測を困難にするものである。

【0015】

【課題を解決するための手段】ICカード用チップに代表される耐タンパ装置は、プログラムを格納するプログラム格納部と、データを保存するデータ格納部を持つ記憶

装置と、プログラムに従い、所定の処理を実行し、データ処理を行う中央演算装置を持ち、プログラムは、中央演算装置に実行の指示を与える処理命令から構成される一つ以上のデータ処理手段からなる、情報処理装置として考えることができる。本発明は、処理しているデータとICカード用チップの消費電流の関連性を減らすための方法として、処理するデータを攪乱用データで変形し、データの処理を変形したデータで処理し、処理後、攪乱用データを使って逆変形し、正しい処理結果を求めるものである。データ処理の後に使用する攪乱データは、必要があれば、データ処理と同じ処理を行った処理済の攪乱用データを使用する。また、攪乱用データは、データ処理ごとにランダムに設定する。このようにすることにより、データ処理において、本来のデータを使わず、毎回異なった変形したデータを使うことができる。そのため、電流波形からデータを推定することが困難にする。

【0016】具体的には、まず、攪乱用データXiを作成し、データD1を変形し、変形データH1を作成する。変形の方法には、排他的論理和や加算、乗算などがある。データ処理では、変形データH1を使い、データ処理を行い、処理済み変形データH2を生成する。本来は、データD1を使用するのであるが、ここでは、変形データH1を使用しているため、変形データH1を処理する電流波形からデータD1のデータを推定することは困難である。変形データH1は、処理ごとに異なる攪乱用データXiにて、変形されているため、毎回、異なるデータとなっている。そのため、変形データH1を処理する電流波形も、毎回、異なった波形となり、その波形から変形データH1を推定しても意味がないことになる。

【0017】攪乱用データXiもデータD1と同じようにデータ処理をする必要がある場合は、攪乱用データXiを入力とし、データ処理を行い、処理済み攪乱用データXoを作成する。そして、処理済み攪乱用データXoを使って処理済み変形データH2を処理し、入力データD1を入力データ処理手段で処理した結果である処理済みデータD2を得る。

【0018】データ変形の方法が異なったものを使用する必要がある場合などでは、データ変形を連結して、実行する必要がある場合がある。その場合は、データ変形、変形データ処理、攪乱用データ処理、データ逆変形処理を入れ子にしたり、組み合わせることにより、本来のデータをデータ処理しないようにデータ変形を連続させる。

【0019】本発明は、暗号アルゴリズムでの、データを入れ替える転置や換字、あるいは、テーブルのアクセスなどの情報隠蔽に使うことができる。データの入れ替えの場合は、排他的論理和をデータ変形に使い、データ処理において、変形したデータと攪乱用データを同じように処理し、常に、変形したデータと攪乱用データを排他的論理和を実行すると、本来のデータを得ることがで

きるように変形処理を行うことが、有効な方法の一つである。

【0020】

【発明の実施の形態】以下、本発明の実施例について図面を参照しながら説明する。

【0021】図1はICカードの外観を示したものである。ICカード101は、ISO7816の規格により大きさや、ICカードのチップ102の位置や接点の数および割り当てなどが規定されている。

【0022】図2は、ICカード用チップ102の内部構成である。構成については、従来技術の説明で既に述べたとおりである。本発明は、プログラム205で処理するデータに攪乱を加えることにより、処理中に生じるICカード用チップのハードウェアが消費する電流波形から、本物のデータの推定を困難にさせるものである。

【0023】基本的な考えを次の簡単な命令列を例にして説明する。

【0024】

logica_shiftl R1 (式2)

xor R1 R2 (式3)

式2は、レジスタR1の値を左に論理的にループさせる命令である。最上位ビットが最下位ビットに移動する。その結果はレジスタR1に格納される。その結果とレジスタR2の内容を排他的論理和し、その結果がレジスタR2に格納される命令列である。このような命令は、DESのような暗号アルゴリズムでは、頻繁に出現する。式2と式3では、処理するデータをそのまま扱っているのでデータの内容によって電流波形の大きさを変える。そのため、電流波形を観測することによりデータを推定できる可能性がある。

【0025】式2と式3の命令で処理するデータを直接扱わないように、以下のような命令列に変更する。

【0026】

xor X1 R1 (式4)

xor X2 R2 (式5)

logica_shiftl R1 (式6)

xor R1 R2 (式7)

logica_shiftl X1 (式8)

xor X1 X2 (式9)

xor X2 R2 (式10)

ここで、X1とX2は勝手に選んだ乱数であり、攪乱用データである。式4と式5では、レジスタR1とR2に乱数X1とX2を排他的論理和を行い、攪乱用データを使い、本物のデータの変形を行う変形処理である。式6と式7は式2と式3と同じデータ処理であるが、変形処理を行ったため、R1とR2の値は、本物のデータの値とは異なったデータとなっている。式8と式9は、攪乱用データ自体をデータ処理を行っているものである。式10で、攪乱用データの処理結果と式7の処理結果を排

Output(j) = f(Input(i))

他論理和することにより、本物のデータにもどす逆変形処理である。これは、請求項7の実施例である。

【0027】これを具体的な数値で値の変化を示すと以下ようになる。R1とR2を以下の値とする。

05 【0028】

R1:11001010 (式11)

R2:01010111 (式12)

式2の結果のR1の値は以下ようになる。

【0029】

10 R1:10010101 (式13)

そして、式3の処理結果は、以下ようになる。

【0030】

R2:11000010 (式14)

次に、本発明の変形を行った場合について示す。まず、攪乱用データを以下のようにする。レジスタR1とR2の値は同じものとする。

【0031】

X1:10010111 (式15)

X2:00111010 (式16)

20 式4と式5を処理した結果、レジスタR1とR2の値は以下ようになる。

【0032】

R1:01011101 (式17)

R2:01101101 (式18)

25 式6と式7を処理した結果のレジスタR1とR2の値は以下ようになる。

【0033】

R1:10111010 (式19)

R2:11010111 (式20)

30 式8と式9で攪乱用データX1、X2も同じようにデータ処理した結果は、以下ようになる。

【0034】

X1:00101111 (式21)

X2:00010101 (式22)

35 そして式10の逆変形処理した結果、本物のデータをそのまま扱った処理結果式14と同じ結果が得られる。

【0035】

R2:11000010 (式23)

この例で示したように、変形処理を行い、変形したデータも攪乱用データも同じデータ処理を行い、その結果を使って逆変形することにより、元の値を求めることができる。そして、データ処理では、本物のデータは使われていないため、その電流波形を見ても処理している変形データは推定できるが、本物のデータを推定することは困難である。

【0036】上で具体的な数値で示した例をより一般的な形式で示す。実際の処理を以下の通りとする。

【0037】

(式24)

これは、 i 個の入力Inputを入力し、処理 f を行い、 j 個の出力Outputを出力する処理である。式2と3の例では、 $R1$ と $R2$ の二つの入力であり、出力は一つで、 $R2$ に格納されていた例である。式24の処理中の電流波

$$\text{InputX}(i) = h(\text{Input}(i), X(i))$$

$$\text{OutputX}(j) = f(\text{inputX}(i))$$

$$\text{Xoutput}(j) = f(X(i))$$

$$\text{Output}(j) = g(\text{OutputX}(i), \text{Xoutput}(i))$$

(式25)

(式26)

(式27)

(式28)

式25は、攪乱用データ $X(i)$ を使って、入力データ $\text{Input}(i)$ を変形し、変形した入力データ $\text{InputX}(i)$ を作成する変形処理である。 h は変形操作である。式26は、変形した入力データを使いデータ処理を行うデータ処理である。式27は攪乱用データを入力データと同じように処理する攪乱用データ処理である、式28は、変形した入力データのデータ処理結果 $\text{OutputX}(j)$ と攪乱用データの処理結果 $\text{Xoutput}(j)$ を変形の逆変換を行う逆変形処理である。 g は逆変形操作である。

【0039】先の例では、式4と5が式25の処理に対応し、変形操作 h は排他的論理和である。式6と式7は式26の変形された入力データのデータ処理に対応する。式8と式9は、式27の攪乱用データのデータ処理

$$\text{Output} = \text{Input}(1) + \text{Input}(2) - \text{Input}(3)$$

(式29)

のような操作は、以下のような変形処理を行い、

$$\text{InputX}(1) = \text{Input}(1) + X(1)$$

(式30)

$$\text{InputX}(2) = \text{Input}(2) + X(2)$$

(式31)

$$\text{InputX}(3) = \text{Input}(3) + X(3)$$

(式32)

変形した入力データを処理することにより変形した入力データの処理結果を得ることができる。

$$\text{OutputX} = \text{InputX}(1) + \text{InputX}(2) - \text{InputX}(3)$$

(式33)

攪乱用データの同じようにデータ処理を行う。

$$\text{Xoutput} = X(1) + X(2) - X(3)$$

(式34)

そして逆変形処理を行う。

$$\text{Output} = g(\text{Xoutput}) = \text{OutputX} - \text{Xoutput}$$

$$= \text{InputX}(1) + \text{InputX}(2) - \text{InputX}(3) - (X(1) + X(2) - X(3))$$

$$= \text{Input}(1) + \text{Input}(2) - \text{Input}(3)$$

(式35)

正しい出力データを求めることができる。これは、加減算の演算では、ある値を加算して演算し、最期の結果に加算した値を減算すれば、正しい結果が求まることからである。これは、請求項8と9の実施例である。

【0044】データ処理 f が乗除算であれば、変形操作 h に乗算、あるいは除算をえらび、逆変形操作に除算、あるいは乗算を選ぶことにより、変形処理と逆変形処理を実現することができる。この理由は、加減算と同じで

$$\text{Output} = \text{Input}(1) + \text{Input}(2) - \text{Input}(3) \bmod N$$

(式36)

入力 $\text{Input}(i)$ を以下のように変形する。

$$\text{InputX}(i) = \text{Input}(i) + k(i) * N$$

(式37)

そして、変形した入力データを使い、加減剰余演算を行う。

$$\text{OutputX} = \text{InputX}(1) + \text{InputX}(2) - \text{InputX}(3) \bmod N$$

(式38)

この式は、式37を使うことにより

形から処理されるデータを推定することを困難にするために、以下のように行う。

【0038】

である。式10は、式28の逆変形処理である。変形操作 g は排他的論理和である。変形操作 h と逆変形操作 g を何に選ぶかはデータ処理 f の特性で決まる。式2と3の処理では、排他的論理和が、変形操作 h であり、逆変形操作 g でもある。これは、シフト操作やXOR操作などは、排他的論理和を変形操作 h として選ぶことにより、逆変形操作 g も排他的論理和を選ぶことができる。排他的論理和は、同じデータを排他的論理和を取ったならば、論理的ゼロになり、排他的論理和を取った操作が消えるからである。

【0040】データ処理 f が加減算であれば、変形操作 h に加算、あるいは減算をえらび、逆変形操作に減算、あるいは加算を選ぶことができる。例えば、

【0041】

【0042】

【0043】

【0045】データ処理 f が剰余演算の加減算であれば、変形操作 h に法 N の整数倍の加減算を選ぶことができる。例えば、以下のような加減剰余演算を考える。

【0046】

【0047】

【0048】

$$\text{OutputX} = (\text{Input}(1) + k(1)*N) + (\text{Input}(2) + k(2)*N) - (\text{Input}(3) + k(3)*N) \text{ mod } N$$

$$= (\text{Input}(1) + \text{Input}(2) - \text{Input}(3)) + (k(1)*N + k(2)*N - k(3)*N) \text{ mod } N \quad (\text{式 } 39)$$

と変形できる。そして、剰余演算の特徴である

$$0 = k * N \text{ mod } N \quad (\text{式 } 40)$$

を使うことにより、式39の2番目の括弧内の値は0となり、式39は、

$$\text{OutputX} = (\text{Input}(1) + \text{Input}(2) - \text{Input}(3)) \text{ mod } N \quad (\text{式 } 41)$$

となる。すなわち、変形した入力データの演算結果は、10 処理が不要である例である。これは、攪乱データのデータ処理の結果が本来の演算結果と同じである。これは、剰余演算の特徴

を使うことにより、攪乱用データのデータ処理と逆変形

$$\text{Xoutput}(i) = k(1) * N + k(2) * N - k(3) * N \text{ mod } N \quad (\text{式 } 42)$$

が、そもそも0になるため、攪乱用データのデータ処理と逆変形処理が不要であるためである。これは、請求項12の実施例である。変形操作hに法Nの整数倍に1を加えたものを使うことができる。例えば、以下のような乗算剰余演算を考え

【0049】データ処理fが剰余演算の乗算であれば、【0050】

$$\text{Output} = \text{Input}(1) * \text{Input}(2) * \text{Input}(3) \text{ mod } N \quad (\text{式 } 43)$$

入力Input(i)を以下のように変形する。【0051】

$$\text{InputX}(i) = \text{Input}(i) + k(i) * N + 1 \quad (\text{式 } 44)$$

そして、変形した入力データを使い、乗算剰余演算を行う。【0052】

$$\text{OutputX} = \text{InputX}(1) * \text{InputX}(2) * \text{InputX}(3) \text{ mod } N \quad (\text{式 } 45)$$

この式は、式44を使うことにより

$$\begin{aligned} \text{OutputX} &= (\text{Input}(1) * (k(1)*N + 1)) + (\text{Input}(2) * (k(2)*N + 1) - (\text{Input}(3) + (k(4)*N + 1)) \text{ mod } N \\ &= (\text{Input}(1) * \text{Input}(2) * \text{Input}(3)) * ((k(1)*N + 1) * (k(2)*N + 1) * (k(3)*N + 1)) \text{ mod } N \end{aligned} \quad (\text{式 } 46)$$

と変形できる。そして、剰余演算の特徴である

$$0 = k * N \text{ mod } N \quad (\text{式 } 47)$$

を使うことにより、式46は、

$$\begin{aligned} \text{OutputX} &= (\text{Input}(1) * \text{Input}(2) * \text{Input}(3)) * (1 * 1 * 1) \text{ mod } N \\ &= \text{Input}(1) * \text{Input}(2) * \text{Input}(3) \text{ mod } N \end{aligned} \quad (\text{式 } 48)$$

となる。すなわち、変形した入力データの演算結果は、【0054】

本物の演算結果と同じである。これも、剰余演算の特徴 35 $1 = X * Y \text{ mod } N \quad (\text{式 } 49)$

を使うことにより攪乱データのデータ処理と逆変形処理が不要である例である。これは、請求項22の実施例である。このような数の簡単なものとして、Xが2で、Yが(N+1)/2であるものがある。これは、変形操作としてXを掛けておき、逆変形でXを掛けた回数だけYを掛けることにより、本物の処理結果を求めるものである。例え

【0053】また、データ処理fが剰余演算の乗算であれば、変形操作hにある数Xと法Nに対して逆数である 40 ば、以下のような加減剰余演算を考える。

$$\text{Output} = \text{Input}(1) * \text{Input}(2) * \text{Input}(3) \text{ mod } N \quad (\text{式 } 50)$$

入力Input(i)を以下のように変形する。【0056】

$$\text{InputX}(i) = \text{Input}(i) * X \quad (\text{式 } 51)$$

そして、変形した入力データを使い、加減剰余演算を行う。 45 【0057】

$$\text{OutputX} = \text{InputX}(1) * \text{InputX}(2) * \text{InputX}(3) \text{ mod } N \quad (\text{式 } 52)$$

そして、逆変形処理gとして、Xを掛けた回数だけYを掛けることにより、正しい結果をえることができる。

けたものを選ぶと、OutputXに逆変形処理gを操作する 【0058】

$$\text{Output} = \text{OutputX} * Y * Y * Y \text{ mod } N$$

$$\begin{aligned}
&= \text{InputX}(1) * \text{InputX}(2) * \text{InputX}(3) * Y * Y * Y \bmod N \\
&= \text{Input}(1) * X * \text{Input}(2) * X * \text{Input}(3) * X * Y * Y * Y \bmod N \\
&= \text{Input}(1) * \text{Input}(2) * \text{Input}(3) * X * X * X * Y * Y * Y \bmod N \\
&= \text{Input}(1) * \text{Input}(2) * \text{Input}(3) * X * Y * X * Y * X * Y \bmod N \\
&= \text{Input}(1) * \text{Input}(2) * \text{Input}(3) \bmod N \quad (\text{式53})
\end{aligned}$$

式53では、式49の性質を使った。この例では、攪乱用データのデータ処理は不要であるが、攪乱用データを掛けた回数だけ、法Nでその逆数を逆変形処理で掛ければ、正しい結果を得ることができる。これは、請求項13と14の実施例である。

【0059】これまでは、データ処理fがデータ操作である例であったが、テーブルからのデータ取り出し操作を攪乱することも電流波形からデータを推定させないために必要である。テーブルのデータの攪乱とテーブルのアドレスの攪乱に関する実施例について、図33のテーブルからデータを取り出す例を用いて説明する。

【0060】図33のテーブルのデータを攪乱用データX1で排他的論理和をとる。ここで攪乱用データとして“9”を選び、テーブルの値と9との排他的論理和を実行する。その結果が図34のテーブルである。次に、テーブルのアドレスを攪乱するために、行番号と攪乱用データX2として選んだ“3”との排他的論理和を実行し、

$$\text{Gyou} = \text{GyouY1} \text{ xor } Y1$$

$$\text{Retsu} = \text{RetsuY2} \text{ xor } Y2$$

しかし、この逆変形処理を行って図33のテーブルを使うことは、真のアドレスデータを使うため、電流波形からアドレスデータを推定される可能性がある。そこで、まず、図35のテーブルを作った時に、行番号と列番号

$$\text{GyouY1X2} = \text{GyouY1} \text{ xor } X2$$

$$\text{RetsuY1X3} = \text{RetsuY2} \text{ xor } X3$$

そして、今まで使ってきた攪乱用データの逆変形処理を行う。

$$\text{GyouX2} = \text{GyouY1X2} \text{ xor } Y1$$

$$\text{RetsuX3} = \text{RetsuY2X3} \text{ xor } Y2$$

こうすることにより、真の行番号や列番号が使われていないため、電流波形からその値を推定することは困難である。GyouX2とRetsuX3を使って、図35のテーブルTab

$$\text{DataX1} = \text{TableX1X2X3}(\text{GyouX2}, \text{RetsuX3})$$

図35のテーブルはすでに攪乱用データX1で変形されているため、これ以降は、攪乱用データをX1として処理を行う。式56から式60までの処理で真のデータを使用していない。これは、請求項16と17の実施例である。

【0066】一定の処理ごとに、テーブルのデータの攪乱用データX1や行番号の攪乱用データX2、列番号の攪乱用データX3をランダムに生成し、テーブルを変形しておく。こうすることにより、処理ごとにテーブルが変形されるため、電流波形からデータを推定することは困難である。

【0067】以上攪乱用データの種類とデータの変形の

列番号と攪乱用データX3として選んだ“2”との排他的論理和を実行し、テーブルを並び替える。その結果が図35である。元のテーブルである図33のテーブルの1行2列目のデータ3301である“0”は、攪乱用データX1と排他的論理和を実行した結果、図34のテーブルでは“9”(3401)となる。そして、行番号と列番号を攪乱用データX2と攪乱用データX3とそれぞれ排他的論理和を実行した結果、3401は、3501の場所に移る。テーブルからのデータ取り出し操作を攪乱するために、このようなテーブルを作成しておく。

【0061】アドレス計算までに、すでに、攪乱用データY1とY2でそれぞれ、行番号変数Gyouと列番号変数Retsuが排他的論理和で変形されているとする。すなわち、正しい行番号Gyouと列番号RetsuはY1とY2をGyouとRetsuに排他的論理和を実行しないと得られないとする。それは、以下のような関係である。

$$\text{【0062】}$$

$$(\text{式54})$$

$$(\text{式55})$$

を攪乱するために使用した攪乱用データX2と攪乱用データX3を用いる。

$$\text{【0063】}$$

$$(\text{式56})$$

$$(\text{式57})$$

$$\text{【0064】}$$

$$(\text{式58})$$

$$(\text{式59})$$

leX1X2X3を参照しDataX1を取る出す。

$$\text{【0065】}$$

$$(\text{式60})$$

方法について説明してきた。次に、その処理手順を示す。図4に基本的な攪乱用データを使った情報隠蔽手順の実施例について示す。

【0068】図4は、基本的な手順である。攪乱用データ生成手段で攪乱用データXiを生成する(401)。この一般的な方法としては、乱数発生器や擬似乱数を使って必要な長さの乱数を生成する方法がある。次に、データ変形処理手段(406)で、入力データD1を攪乱用データXi(405)で変形し、変形データH1(407)を生成する。変形方法は、先に述べたように、排他的論理和や、加減算、乗除算などがある。そして変形データH1を使ってデータ処理を変形データ処理手段(408)

で行い、処理済み変形データ H2 を生成する。一方、攪乱用データ X_i は入力データと同じデータ処理を攪乱用データ処理手段(403)で行い、処理済みの攪乱用データ X_o(404)を作成する。そして、攪乱用データ X_o と処理済み変形データ H2 を使い、データ逆変形処理手段(410)で真の処理済みデータ D2(411)を求める。データ変形処理手段(406)やデータ逆変形処理手段(410)の方法には、すでに説明した、排他的論理和や加減算、乗除算、剰余演算などがある。これは、請求項1の実施例である。

【0069】図5の実施例は、攪乱用データを二つ使った例であり、第1の攪乱用データを使った情報隠蔽処理の中に第2の攪乱用データを使った情報隠蔽処理が含まれている場合である。主な流れは、図4の実施例と同じ

$$\begin{aligned} H1 &= D1 \text{ xor } X1i \\ H2 &= f1(H1) \\ X1o &= f1(X1i) \\ H31 &= H2 \text{ xor } X2i \\ H32 &= D2 \text{ xor } X2i \\ H4 &= f2(H31, H32) \\ X2o &= f2(X2i, X2i) \\ H5 &= H4 \text{ xor } X2o \\ D2 &= H5 \text{ xor } X1o \end{aligned}$$

ここで f1 と f2 はデータ処理操作である。この例のように、第2のデータ処理 f2 で、別のデータ D2 を使う場合で、そのデータ D2 を第2の攪乱用データで変形して使用する場合に、本実施例の処理手順を使うと有効である。これは、請求項2の実施例である。

【0071】図6の実施例も、攪乱用データを二つ使った例である。図5の実施例との大きな違いは、第1の攪乱用データを使った情報隠蔽処理と第2の攪乱用データを使った情報隠蔽処理がつながっている場合である。そして、第1の攪乱用データによる変換を戻す逆変形処理の前に、第2の攪乱用データで変形しておき、真のデータを使った処理を隠蔽するための手順である。第1の攪乱用データで変形された変形データ H1(609)は、

$$\begin{aligned} H1 &= D1 \text{ xor } X1o \\ H2 &= f1(H1) \\ X1o &= f1(X1i) \\ H3 &= H2 \text{ xor } X2i \\ H4 &= H3 \text{ xor } X1o \\ H5 &= f2(H4) \\ X2o &= f2(X2i) \\ D2 &= H5 \text{ xor } X2i \end{aligned}$$

これは、処理操作が複数あり、攪乱用データを複数使う必要な場合に有効である。これは、請求項3の実施例である。

【0073】図7の実施例は、攪乱用データの面からは、データ処理をあらかじめ計算しておき、処理を効率化するものである。あらかじめ、攪乱用データ処理手段

である。第1の攪乱用データで変形された変形データ H1(507)をデータ処理し、その処理結果である処理済み変形データ H2(509)に第2の攪乱用データ X2_i を使って第2データ変形処理手段(510)で変形を行い、処理済み変形データ H3(511)を作成する。そのデータを第2変形データ処理手段(512)でデータ処理を行い、処理済み変形データ H4 を生成し、第2の攪乱用データの逆変換を第2データ逆変形処理手段(520)でおこない、処理済み変形データ H5(521)を生成する。そして、第1の攪乱用データの逆変換を第1データ逆変形処理手段(514)で行い、真の処理済みデータ D2(515)を求める。変形に排他的論理和を使った例は、以下の通りである。

【0070】

(式61)

第1変形データ処理手段(610)でデータ処理を行い、その処理済み変形データ H2(611)を第2変形データ作成手段(612)で第2の攪乱用データを使って変形し、処理済み変形データ H3(613)を生成する。そして、第1攪乱用データの逆変換を第1データ逆変形処理手段(605)で行う。その処理結果である処理済み変形データ H4(606)を使って第2変形データ処理手段(614)を行い、処理済み変形データ H5(615)を生成する。そして、第2データ逆変形処理手段(616)で、逆変換を行い、真の処理済みデータ D2(617)を生成する。変形に排他的論理和を使った例は、以下の通りである。

【0072】

(式62)

で処理済み攪乱用データ X_o を生成し(703)、処理済み攪乱用データ記録手段(706)で、記録しておく。処理の中では、データ逆変形処理手段(713)では、記録しておいた処理済み攪乱用データ(714)を読みだし、使用する。これは、同じようなデータ処理を何回も実行する場合には効率の面で有効である。しかし、攪

乱用データが何回も使われるため、情報の隠蔽性にとって、図4の実施例のように攪乱用データを毎回変更する方が、有効である。これは、処理速度と情報の隠蔽性のトレードオフから決まる。これは、請求項5の実施例である。

【0074】図8の実施例は、第1と第2の攪乱用データの逆変形を統合して行い、その後その結果を使って、データの逆変形を行うものである。第1と第2の攪乱用データは、それぞれ、第1攪乱用データ処理手段(803)と第2攪乱用データ処理手段(807)で処理済み攪乱用データX1oとX2oが作られる。それらのデータをデータ逆変形統合手段で統合し、統合処理済み攪乱用データXoを生成する。そのデータを使い、第1変形データ処理(814)及び第2変形データ処理(818)を処理した結果である処理済み変形データH4(819)の逆変形処理手段(820)を行い、真の処理済みデータD2を生成する。これは、個別に逆変形する場合よりも、処理済み攪乱用データを統合し、そのあと、まとめて変形処理を行う方法である。逆変形処理に処理時間がかかる場合に有効である。これは、請求項4の実施例である。

【0075】次に、対称暗号DES(data encryption standard)を例にした実施例について説明する。本発明は、他の暗号にも使うことができる。暗号については、共立出版株式会社岡本栄司著「暗号理論入門」などに、記載されている。

【0076】DESは、64ビットのデータ(平文か暗号文)を56ビットの暗号鍵で、暗号化と復号化を行う。暗号化と復号化で同じ暗号鍵を使用するため、対称暗号と呼ばれている。DESは、トランプをきって、ランダムにするように、平分(暗号化される文)のビットをランダムに入れ替え暗号化する。データの入れ替えは、暗号鍵にしたがって行う。復号化する時は、暗号鍵に従い入れ替えた順と逆に入れ替えを行い、データを元に戻す。DESの処理での入れ替えは、ビット単位と複数ビットまとまった単位での二つの入れ替え方法がある。前者は、転置と呼び、後者は、換字と呼ぶ。

【0077】図9を用いて、DESの暗号化の方法を説明する。変形処理a(901)と変形処理b(904)、逆変形処理(916)は、本発明に関するものであり、DESの本来の暗号処理には関係がない。暗号文は、まず、初期転置(IP: Initial Permutation)902で転置させられる。これは、初期転置テーブルに従い、ビット単位で、暗号文の64ビットのデータを入れ替える。これ以降、初期転置の逆転置(IP-1)916まで、一組の操作を16段行う。

【0078】各1段の処理は、前段の結果の前半か後半の32ビットのデータと暗号鍵を入力としてf関数903と呼ばれる処理を行い、その出力を前段の残りの半分のビットを使って排他的論理和909を取る操作を行う。暗号鍵も、入れ替えが行われる。暗号鍵に対して、

まず、PC-1というテーブルを使った選択転置PC-1(905)が行われる。その後、PC-2というテーブルを使った選択転置PC-2(908)を行い、暗号鍵の入れ替えを行う。次の段では、28ビットづつをLSテーブルにしたがって巡回させて使用する。

【0079】本実施例では、IP処理の前に、平文を変形するための変形処理a(901)と暗号鍵を変形するための変形処理b(904)、そして、最後に逆変形する処理(916)を追加する。変形処理a(901)は、平文を変形し、IP処理(902)やf関数(903)の処理で、平文そのものを処理するのではなく、変形した平文を処理することにより、その処理の電流波形から平文のデータを推定できないようにするものである。変形処理b(904)は、PC-1処理(905)やLS処理(907)、PC-2処理(908)、f関数(903)で暗号鍵そのものを処理するのではなく変形した暗号鍵を処理することにより、その処理の電流波形から暗号鍵の推定を困難にするものである。

【0080】f関数303の処理を、図10に示す。まず、f関数への入力文を選択的転置行列Eに基づいて、選択的転置を行う(1002)。次に、選択的転置を行った入力データと暗号鍵との排他的論理和をとり(1003)、Sボックスの処理を行い(1004)、P転置処理を行う(1005)。Sボックスの処理は、1003の排他的論理和の処理結果である48ビットから6ビットづつ取り出し、8つのSボックステーブルの行番号と列番号を求め、4ビットのデータを生成する処理である。各Sボックステーブルは、6ビットごとで示されているデータの位置により異なる。P転置処理は、P転置テーブルに従い、32ビットのビット位置を入れ替える操作である。

【0081】変形処理a(901)と変形処理b(902)は基本的に同じ処理である。図11を用いて変形処理aの平文の変形データの作成処理について説明する。攪乱用データX1をランダムに生成する。これは、DESの暗号化(あるいは復号化)処理ごとに乱数発生器か擬似乱数を使って生成する(1102)。毎回、異なった攪乱用データを使用する。次に攪乱用データX1と平文PをXOR(排他的論理和)を実行し、変形平文(変形した平文)PX1を生成する(1103)。DESの場合は、平文は64ビットであるが、生成する乱数は、64ビットでも8ビットでも構わない。しかし、64ビット以下ならば、拡張するようなことを行い、64ビットの攪乱用データX1を生成する必要がある。生成した乱数が8ビットであれば、それを8回繰り返して64ビットの攪乱用データX1を生成してもよい。ここで、排他的論理和(XOR)を使って変形したため、攪乱用データX1と変形平文PX1をXORすると、平文Pが生成される。これは、請求項6の実施例でもある。

【0082】変形処理b(904)の暗号鍵の変形デー

タの作成手順を図36に示す通りである。平文と攪乱用データX1の代わりに暗号鍵Kと攪乱用データX2を用いている点が、図11の実施例と異なるだけである。DESでは、暗号鍵は平文と同じ64ビットである。処理で変形された暗号鍵KX2が生成される。

【0083】次に、IP処理(902)について説明する。IP処理は、図37に示すテーブルにしたがって平文64ビットの並びを入れ替えるものである。テーブルに従い、出力の第1ビットは入力第58ビット、出力の第2ビットは入力第50ビット、出力の第64ビットは入力第7ビットと入れ替える。本実施例でのIP処理を、図12を用いて説明する。まず、変形平文PX1をIP処理し、IP処理済み変形平文PX1IPを生成する(1202)。ビットの入れ替えは、図37のテーブルに従う。次に、攪乱用データX1も同じようにIP処理し、IP処理済み攪乱用データX1IPを生成する(1203)。IP処理済み変形平文PX1IPとIP処理済み攪乱用データX1IPとは排他的論理和を取ると、平文をIP処理した結果を生成することができる。これは、IP処理がビットの移動であるため、攪乱用データも変形平文PX1と同じように移動したため、ビットごとに排他的論理和を実行すると真のデータが求まるという関係が維持されているためである。IP処理の下位32ビットが第1段のf関数(903)と第2段の排他的論理和に使われ、上位32ビットは、排他的論理和(909)の入力となる。これは、請求項1、7、18、20の実施例である。

【0084】IP処理では、変形平文PX1のビットの値は、元の平文のビットの値と異なるため、IP処理の電流波形を見ても、平文のデータを推定することは困難である。“1”のビットの数によって、消費電流は大きくなるが、変形平文での“1”のビットの数は、平文の“1”のビットの数と何ら関係はない。そのため、電流波形の大きさから、平文のデータを推定することは困難である。このように、平文を攪乱用データで変形することにより、処理中の電流波形を観測しても元のデータを推定することが困難にすることができる。

【0085】PC-1処理は、IP処理とほとんど同じである。図38のPC-1用の変換テーブルを使い、64ビットの暗号鍵をバリティビットの8ビットを取り除き、56ビットにし、かつビットの順序を入れ替えるものである。図38のテーブルの見方は図37のテーブルの見かたと同じである。PC-1処理済みの変形暗号鍵KX2PC1とPC-1処理済みの攪乱用データX2PC1は、排他的論理和を実行すると、正しいPC-1処理済みの暗号鍵が求まる。

【0086】LS処理は、PC-1処理で生成された58ビットの鍵を右と左に28ビットずつに分け、それぞれLSテーブルにしたがって左に1ビットあるいは2ビットシフトするものである。実施例を図15を用いて説明する。まず、1502で、PC-1処理済み変形暗号鍵KX2PC1した結果をLS処理し、PC-1及びLS処理済み変形暗

号鍵KX2PC1LSを生成し、1503で、PC-1処理済み攪乱用データX2PC1した結果をLS処理し、PC-1及びLS処理済み攪乱用データX2PC1LSを生成する。LS処理もビット位置の入れ替えであるため、PC-1及びLS処理済み変形暗号鍵KX2PC1LSとPC-1及びLS処理済み攪乱用データX2PC1LSの排他的論理和を実行すると、真のLSを処理した暗号鍵の結果が得られる。LS処理でも、攪乱用データを使ったため、実際に操作されているデータは、真の暗号鍵とは違うため、電流波形を観測しても、暗号鍵の推定は困難である。

【0087】PC-2処理は、LSの処理結果の56ビットをPC-2テーブルにしたがって、48ビットに縮約型転置する。1402で、PC-1及びLS処理済み変形暗号鍵KX2PC1LSをPC-2処理し、PC-1及びLS、PC-2処理済み変形暗号鍵KX2PC1LSPC2を生成する。1403で、PC-1及びLS処理済み攪乱用データX2PC1LSをPC-2処理し、PC-1及びLS、PC-2処理済み攪乱用データX2PC1LSPC2を生成する。基本的には、テーブルを使った転置であるため、PC-1処理と基本的に同じである。

【0088】次に、f関数903の処理について説明する。f関数は、図10に示すように、選択的転置E処理(1002)、暗号鍵と選択的転置の実行結果の排他的論理和(1003)、Sボックス処理(1004)、P転置処理(1005)からなる。

【0089】選択的転置E処理について図16を用いて説明する。選択的転置Eは、IP処理と同じように図28の転置テーブルを使って、ビットの並びを変更するものである。1602で、IP処理済み変形平文PXIPを選択的転置E処理し、IP処理及びE転置処理済み変形平文PXIPEを生成する。そして、1603でIP処理済み攪乱用データXIPを選択的転置E処理し、IP処理及びE転置処理済み攪乱用データXIPEを生成する。IP処理やPC-1処理と同じように、IP処理及びE転置処理済み変形平文PXIPEとIP処理及びE転置処理済み攪乱用データXIPEを排他的論理和をとれば、正しいIP処理及びE転置処理済み平文が求まる。また、転置Eテーブルを使ってビットに入れかを行う際も、ビットの値は、変形されているビットの値であるので、その処理の電流波形を見ても真のデータを推定することは困難である。

【0090】次に、f関数での2番目の処理である暗号鍵と選択的転置の実行結果の排他的論理和の処理を行う。この処理について、図17を用いて説明する。1702では、平文から生成したIP処理及びE転置処理済み変形平文PXIPEと、暗号鍵から生成したPC-1及びLS、PC-2処理済み変形暗号鍵KX2PC1LSPC2とをXORし、Sボックス処理の入力となる48ビットのSボックス入力データSinutXを作成する。次に1703では、平文用の攪乱用データから生成したIP処理及びE転置処理済み攪乱用データXIPEと、暗号鍵用の攪乱用データから生成したPC-1及びLS、PC-2処理済み攪乱用データX2PC1LSPC2とをXO

Rし、Sボックス入力データSinputX用の攪乱データであるSボックス入力データ攪乱用データXSinputを生成する。排他的論理和の性質から、Sボックス入力データ攪乱用データXSinputは、二つの攪乱用データ(PC-1及びLS、PC-2処理済み攪乱用データX2PC1LSPC2と、PC-1及びLS、PC-2処理済み攪乱用データX2PC1LSPC2)をXORするこ

$$PX1 = P \text{ xor } X1$$

$$KX2 = K \text{ xor } X2$$

そうすると、PとKの排他的論理和を実行した結果をZとすると、PX1とPX2の排他的論理和を実行した結果Z1と

$$Z = P \text{ xor } K$$

$$Z1 = PX1 \text{ xor } KX2$$

$$= (P \text{ xor } X1) \text{ xor } (K \text{ xor } X2)$$

$$= P \text{ xor } X1 \text{ xor } K \text{ xor } X2$$

$$= (P \text{ xor } K) \text{ xor } (X1 \text{ xor } X2)$$

$$= Z \text{ xor } (X1 \text{ xor } X2)$$

すなわち、Z1を真のデータに戻すための攪乱用データとしてPとKの攪乱用データを排他的論理和を使えばよいことが分かる。暗号鍵と選択的転置の実行結果の排他的論理和の処理では、Sボックス入力データSinputXの攪乱用データとして、平文用の攪乱用データから生成したIP処理及びE転置処理済み攪乱用データXIPEと、暗号鍵用の攪乱用データから生成したPC-1及びLS、PC-2処理済み攪乱用データX2PC1LSPC2とをXORして生成したSボックス入力データ攪乱用データXSinputを使えばよいことが分かる。これは、請求項18の実施例である。

【0093】次に、Sボックスの処理について図18を用いて説明する。Sボックス入力データSinputXから6ビットずつ取り出し、8つのSボックスについて処理を行う。DESで使われている1番目のSボックスを、図25で示す。8つのSボックスは形式は同じであるが、それぞれフィールドのデータが異なる。それぞれのSボックスの処理では、まず、Sボックス入力データSinputの上位i番目の6ビットのサブデータSubSinputX(i)を取り出す(1805)。前もって作成しておいたSボックスを変形した変形Sボックステーブルのアドレス攪乱用データXsa(i)とSubSinputX(i)をXORし、SubSinputXXsa(i)を生成する(1806)。そのSubSinputXsa(i)をSボックス入力データ攪乱用データXSinputの上位i番目の6ビットとXORし、SubSinputXsa(i)を生成する(1807)。SubSinputXsa(i)は、i番目のSボックスを取り出すための本物のアドレスデータに対して、アドレス攪乱用データXsa(i)が排他的論理和された値である。SubSinputX(i)とXSinput(i)を排他的論理和(XOR)すると、それは、真のデータに戻るなのである。XSinput(i)と排他的論理和する前に、SubSinputX(i)とXsa(i)の排他的論理和を行い、その後で、XSinput(i)を排他的論理和(XOR)を実行する。こうすることにより、本物のデータの処理を行う必要がなくなり、その電流波形からデータを推定することが困難になっている。次に、SubSinpu

とにより生成できる。これを簡単な例で示す。ここ、平文をPと鍵をK、その変形平をPX1、変形鍵をKX2とする。それらの関係は、式63と式64である。X1とX2は、平文と鍵の攪乱用データである。

05 【0091】

$$(式63)$$

$$(式64)$$

Zの関係は、以下のようになる。

10 【0092】

$$(式65)$$

$$(式66)$$

tXsa(i)を使って、変形Sボックステーブルのアドレスを計算する(1808)。本来のSボックスのテーブルを参照するためのアドレスが変形されているため、テーブルをあらかじめ変形しておく必要がある。計算したアドレスを元に、変形SボックステーブルS(i)からSボックス出力データSoutputX3(i)を取り出す(1809)。そして、同時にSボックスの出力データSoutputX3(i)用の攪乱用データX3(i)を取り出す(1810)。

20 8つのSボックスを処理し、SoutputX3(i)とX3(i)をiを1から8までのデータをそれぞれ連結することによって、SoutputX3とX3を生成する。これ以降は、処理データは、SoutputX3となり、攪乱用データはX3となる。これは、請求項21の実施例である。

30 【0094】次に、変形Sボックステーブルの生成方法について図23と図24を用いて説明する。このSボックスで、S(i)ボックス用のアドレス攪乱用データXsa(i)とデータ攪乱用データX3(i)を作成する(2306)。Xsa(i)は6ビットであり、X3(i)は4ビットである。攪乱用データX3は4ビットずつ作成したX3(i)を8個集めた32ビットのデータである。次に、変形S(i)ボックステーブル作成ルーチンと呼びだす(2307)。

35 i番目の変形Sボックステーブル作成ルーチンの処理を図24で説明する。kは行番を指定し、1は列番号を指定する。k行1列の処理は2408から2413である。1番目のSボックスのテーブルは、図25に示す通りである。まず、i番目のオリジナルのSボックスの行番号k列番号1のデータdを取り出す(2408)。

40 そして、そのデータdと攪乱用データX3(i)を排他的論理和を取り、それをd2とする(2409)。これは、攪乱用データを“7”とすると、元のSボックスのデータ2504に対して、その結果は2604になる。これをすべてのフィールドに行うと、図26のようになる。図26のテーブルは、図25の1番目のSボックスのデータに対して、攪乱用データを“7”として排

45 50

他の論理和を取ったものである。

【0095】次に、アドレスの攪乱を行う。まず、Xsa1をXsa(i)の上位1ビットと下位1ビットから作った2ビットとし、Xsa2をXsa(i)の上位2ビットから5ビットから作った4ビットのデータとする。これは、Sボックスのアドレスの計算方法に由来している。そして、図26の行番号と列番号をそれぞれkとlとすると、それぞれにXsa1とXsa2を排他的論理和を行う(2412)。新しくできた行番号と列番号をk2とl2とすると、i番目の変形SボックステーブルS(i)のk2行l2列にデータd2を格納する(2413)。この処理の例を図27に示す。図27は、図26の行と列に対して攪乱用データとして“2”と“9”を選んで作成したものである。わかりやすいように、行と列のデータの位置はそのままにして行と列の番号だけを変更している。図25で、3行1列のデータ12(2504)は、図27では、1行8列に移り、値が11に変形されている。この例では、データの攪乱用データは“7”であり、アドレス用の攪乱データは行が“2”であり、列が“9”である。このようにして8つのSボックスを変形する。本実施例では、この処理は、DESの最初に行っておく。変形したSボックステーブルは、DESの16段で使用する。これは、請求項5の実施例でもある。

【0096】Sボックスの処理が終了すると、処理データは、32ビットのSoutputX3となり、攪乱用データは32ビットのX3となる。これが、f関数の最後の処理である転置P処理(1005)の入力となる。転置P処理を図19を用いて説明する。Sボックスの出力であるSinputX3を転置Pし、SinputX3Pを生成する(1902)。SinputX3の攪乱用データX3を転置Pし、X3Pを生成する(1903)。転置Pに使用するテーブルは図29の通りである。このテーブルの使い方もIP処理のテーブルと同じである。

【0097】f関数の処理が終わると、転置P処理と前段の結果のXORを行う(909とか914)である。これは、まず、Sボックスの結果を転置PしたSinputX3Pと前段の結果をXORする(2002)。そして、X3Pと前段の攪乱用データXをXORする(2003)。このXORの処理は、選択的転置E処理結果と暗号鍵のXOR処理(1701)と同じである。

【0098】DESでは、最後にIP-1処理(915)を行う。この処理を図21に示す。IP-1処理は、IP処理と似たようなビットの並びの入れ替え処理であり、IPテーブルの代わりにIP-1テーブルを使用する(2102)。これまでの処理結果をIP-1処理し、同様に攪乱用データXをIP-1処理する(2103)。

【0099】最後に、正しい処理結果に戻すために、逆変形処理を行う(916)。逆変形処理を図22に示す。IP-1処理結果をIP-1処理した攪乱用データXでXORすることにより、正しい結果を求める。ここで、初め

て、変形されていない正しい処理結果が得られる。

【0100】これまで処理されるデータの隠蔽について述べてきたが、攪乱用データも隠蔽する必要がある場合がある。基本的考え方は、攪乱用データを攪乱用の攪乱用データXRと排他的論理和を取り、変形することである。ただし、XRは、固定にしておき、あらかじめ、ビット位置の入れ替えなどを計算し逆変形のためのXR0を求めておく。そして、攪乱用データが必要になったときに、XR0を使って、元の攪乱用データを求めることにより、攪乱用データの変形と逆変形の実施と効率化を図る。まず、暗号鍵用の攪乱用データを例にして説明する。図30の処理は、暗号鍵用の攪乱用データを攪乱用の攪乱用データXRで排他的論理和をとる変形操作である。変形処理b

(3601)で攪乱用データX2を生成した後に、図30の攪乱用データの変形処理を行う。暗号鍵用の攪乱用データX2はPC-1処理やLS処理、PC-2処理を受ける。これらは、あらかじめ、決まったビットの位置の入れ替えであるので、前もって決まっている値XRに対して、PC-2処理まで実行したときの攪乱用データXR0を求めておく記録しておく(3102から3105)。PC-2処理した後に、1403で生成したPC-1及びLS、PC-2処理済み攪乱用データX2PC1LSPC2に記録しておいた攪乱用データXR0をXORして(3202)、本来のPC-1及びLS、PC-2処理済み攪乱用データX2PC1LSPC2を求めることができる。このようにすることにより、攪乱用データも隠蔽することができる。ここで、攪乱用の攪乱用データXRとその処理済攪乱用データXR0は、同じデータを使ってもよい。

【0101】DESでの実施例は、暗号化であるが、DESのアルゴリズムは、復号化でもほとんど同じであるため、本実施例はほとんど変更しないで適用できる。また、DES以外の暗号アルゴリズムも、転置処理や換字処理、剰余演算を多く使っているため、本発明を適用し、データを変形し、電流波形から本来のデータを推定することを困難にすることができる。

【0102】

【発明の効果】本発明によれば、ICカードチップでの処理データを変形することにより、消費電流の波形から、処理や暗号鍵の推測を困難になる。

【図面の簡単な説明】

【図1】ICカードのハードウェア構成。

【図2】ICカード用チップ内のハードウェア構成。

【図3】消費電流の波形。

【図4】一つの攪乱用データを使ったデータ変形の手順。

【図5】入れ子で二つの攪乱用データを使ったデータ変形の手順。

【図6】連続で二つの攪乱用データを使ったデータ変形の手順。

【図7】攪乱用データのデータ処理をあらかじめ計算しておくデータ変形の手順。

【図 8】二つの攪乱用データの逆変形を統合したデータ変形の手順。

【図 9】DESの全体処理の流れ。

【図 10】DESの f 関数の処理の流れ。

【図 11】変形処理 a。

【図 12】IP処理。

【図 13】PC-1処理。

【図 14】PC-2処理。

【図 15】LS処理。

【図 16】選択的転置E処理。

【図 17】選択的転置E処理結果と暗号鍵のXOR処理。

【図 18】Sボックス処理。

【図 19】転置P処理。

【図 20】転置P処理と前段の結果のXOR処理。

【図 21】IP-1処理。

【図 22】逆変形処理。

【図 23】変形Sボックスのテーブルの作成。

【図 24】i番目の変形Sボックステーブル作成ルーチン。

【図 25】1番目のSボックステーブル。

【図 26】1番目のSボックステーブルのデータを変形したテーブル。

【図 27】1番目のSボックステーブルの配置を変形したテーブル。

【図 28】選択的転置Eテーブル。

【図 29】転置Pテーブル。

【図 30】攪乱用データの暗号化処理。

【図 31】攪乱用データの暗号化用データの変形計算処理。

【図 32】攪乱用データの復号化処理

【図 33】オリジナルテーブルの例。

【図 34】図 33の内容を変形したテーブル。

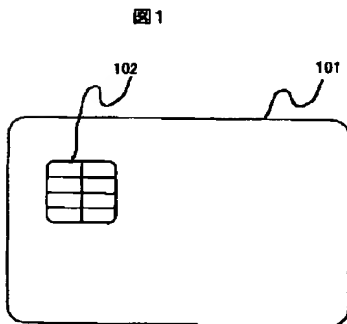
【図 35】図 34のテーブルの配置を変形したテーブル。

【図 36】変形処理 b。

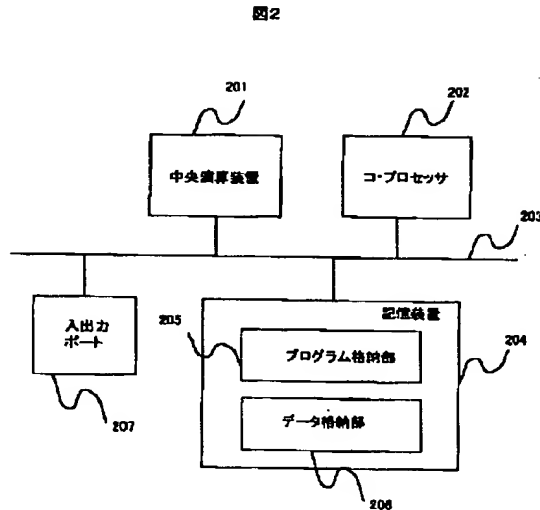
【図 37】IP転置テーブル。

【図 38】PC-1選択的転置テーブル。

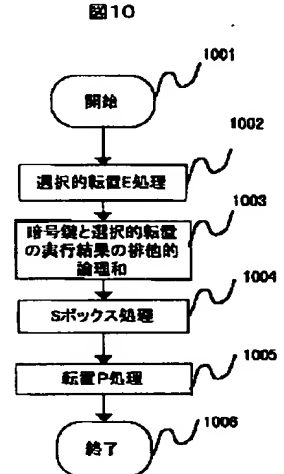
【図 1】



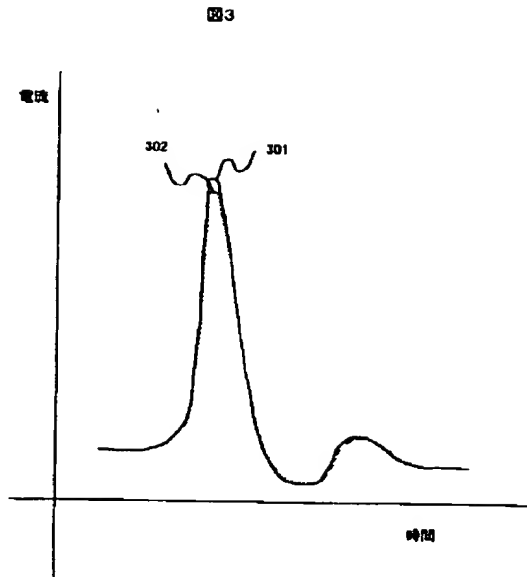
【図 2】



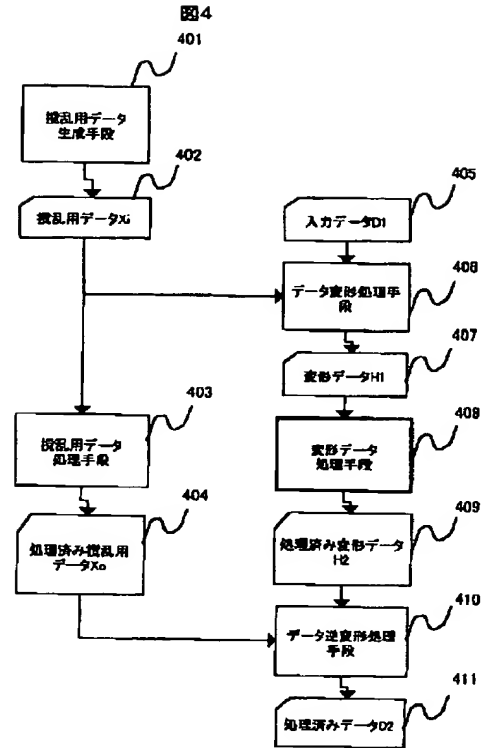
【図 10】



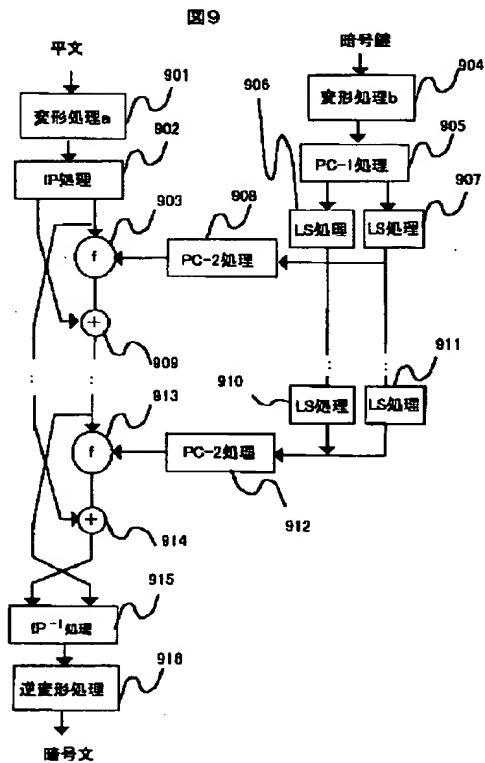
【図3】



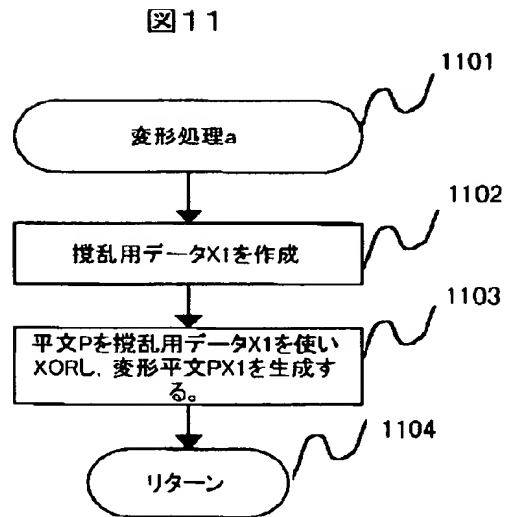
【図4】



【図9】

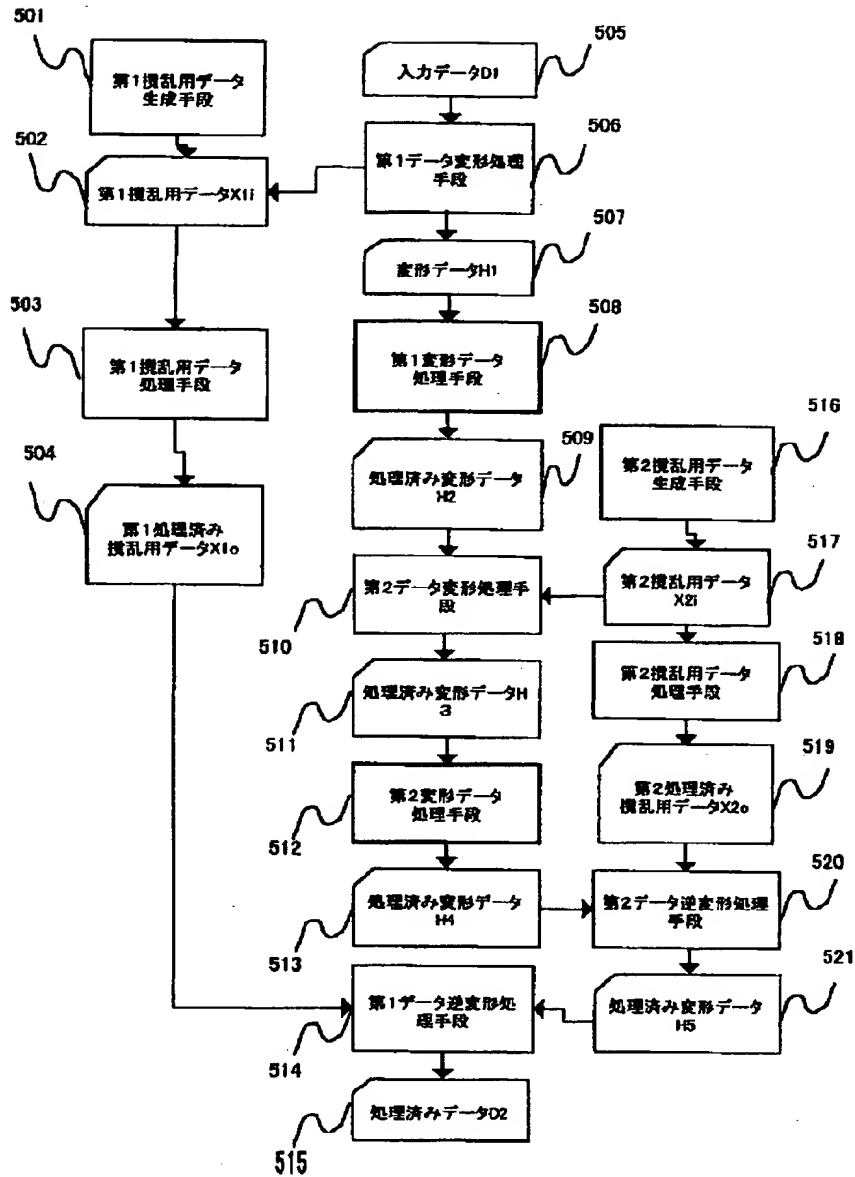


【図11】



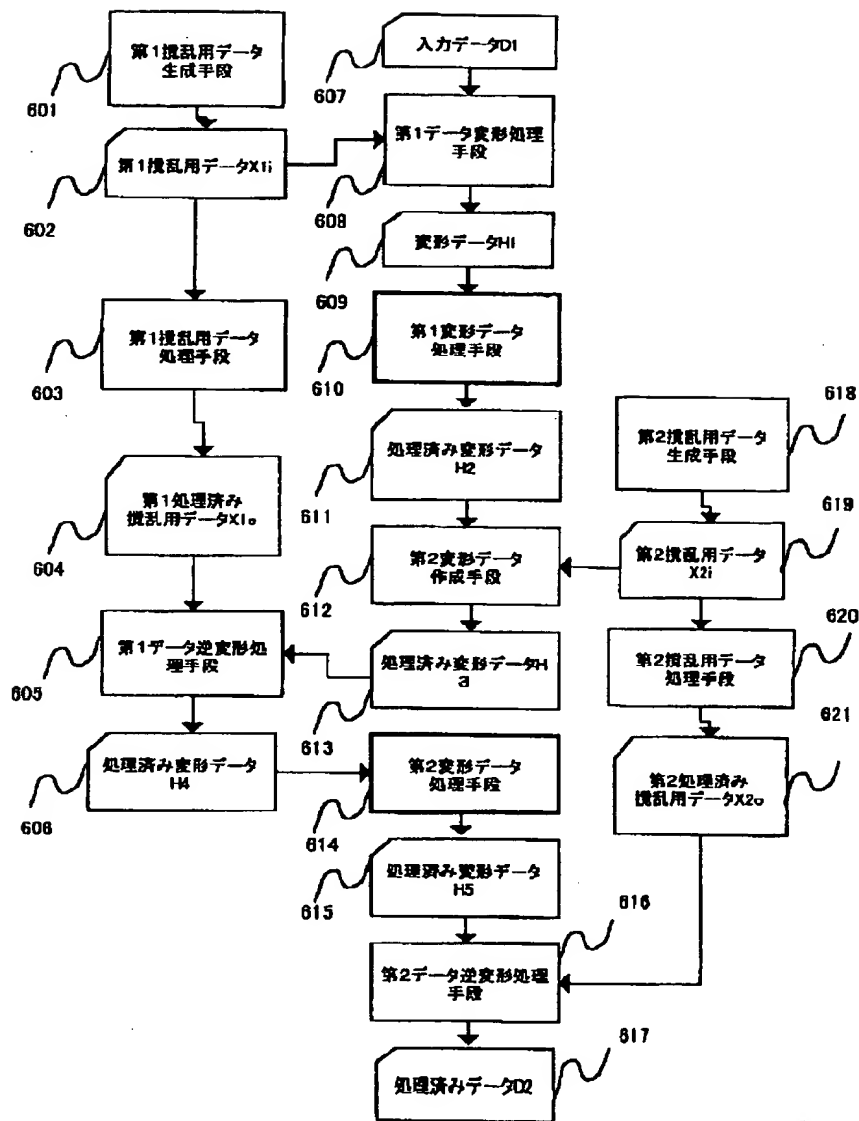
【図5】

図5



【図6】

図6



【図25】

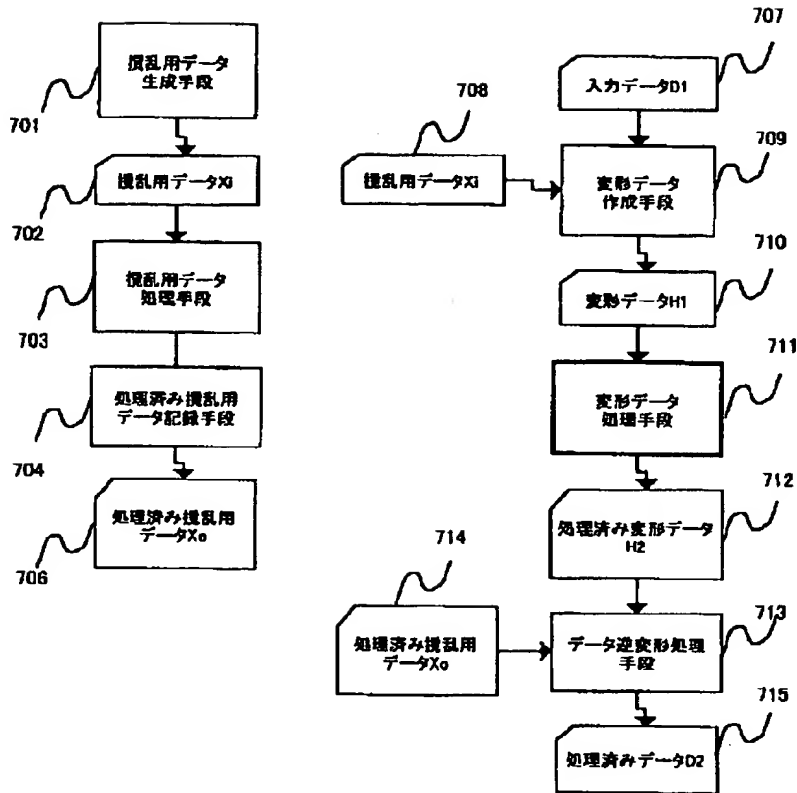
図25

A 6x6 grid of numbers, labeled 2501, 2502, 2503, and 2504. The grid contains the following values:

	0	1	2	3	15
0	14	4	13	1	7
1	0	15	7	4	8
2	4	1	14	8	0
3	15	12	8	2	13

【図7】

図7



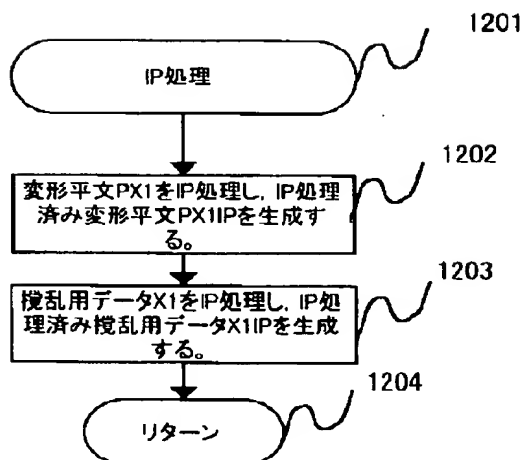
【図29】

図29

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	21	14
32	27	3	9
19	13	30	6
22	11	4	25

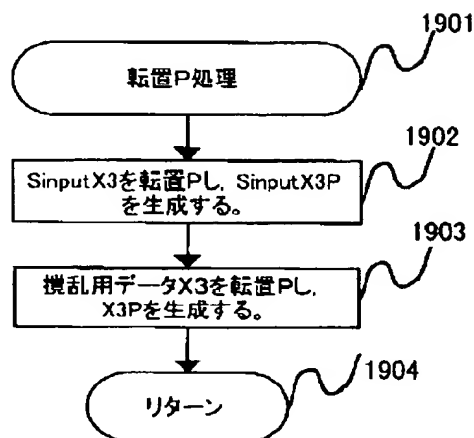
【図12】

図12



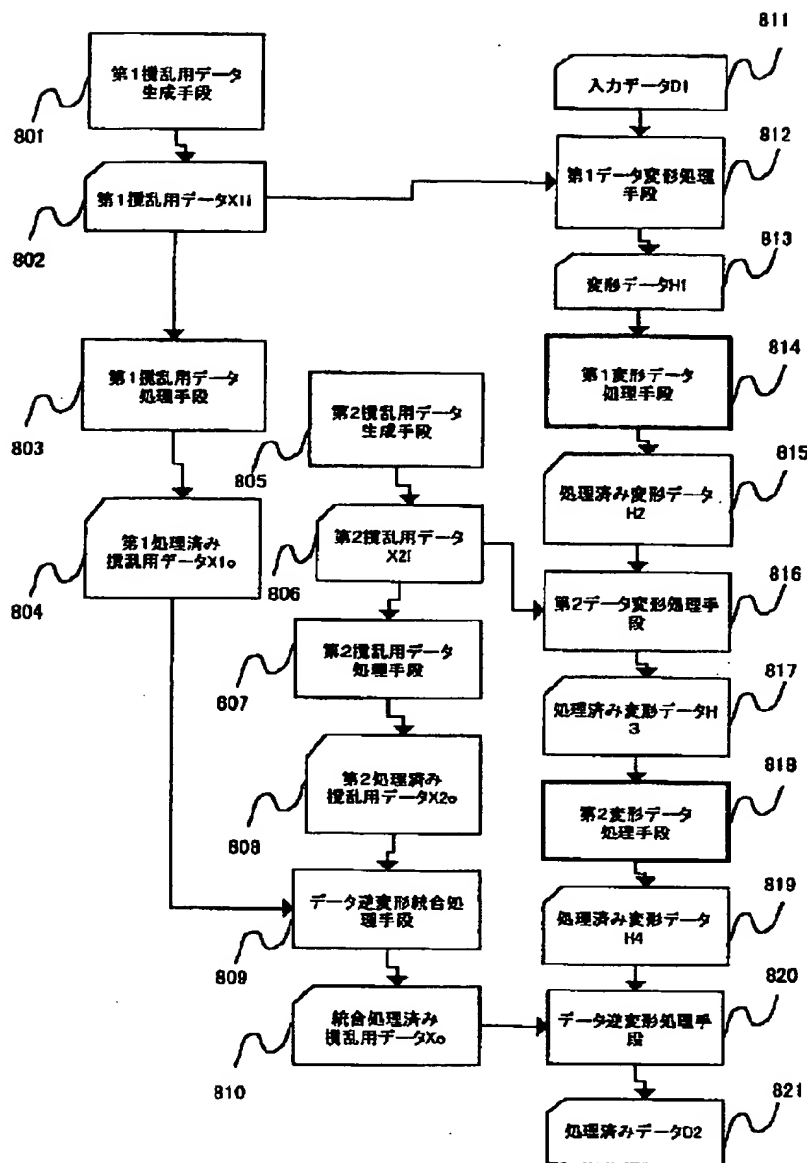
【図19】

図19



【図8】

図8



【図33】

図33

3301

	0	1	2	3
0	B	3	8	7
1	1	4	0	D
2	2	F	A	9
3	6	C	5	E

【図34】

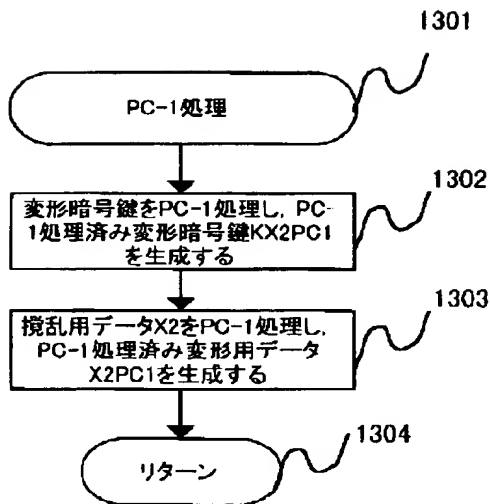
図34

3401

	0	1	2	3
0	2	A	1	E
1	8	D	9	4
2	B	6	3	0
3	F	5	C	7

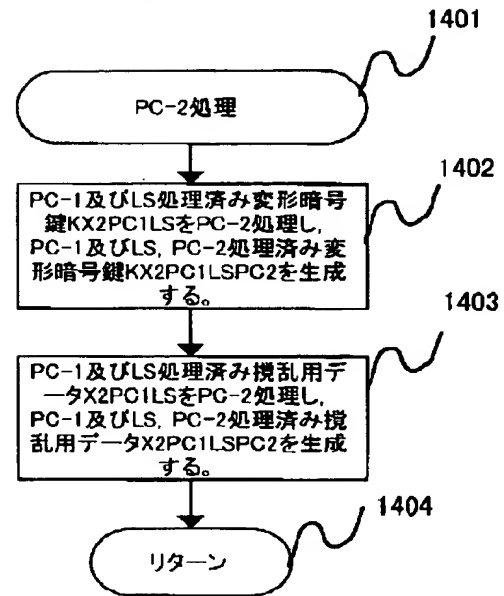
【図13】

図13



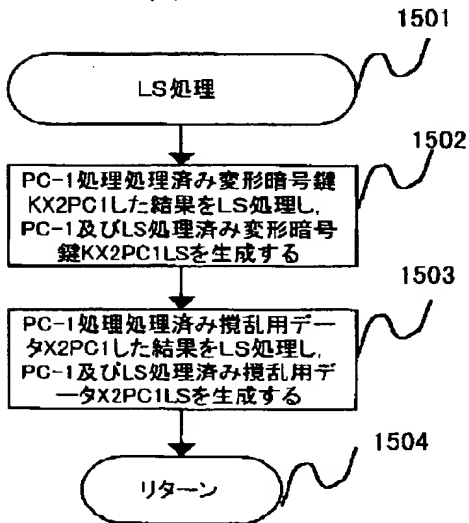
【図14】

図14



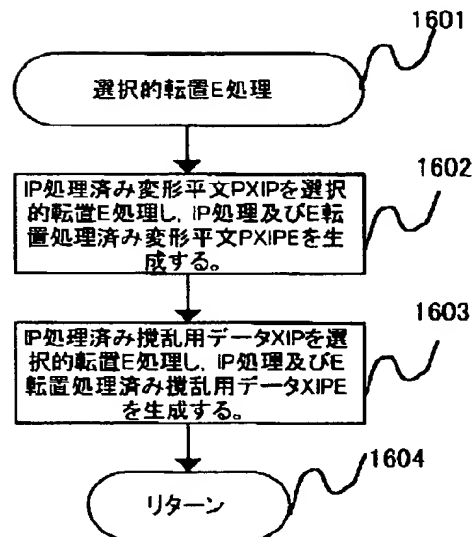
【図15】

図15

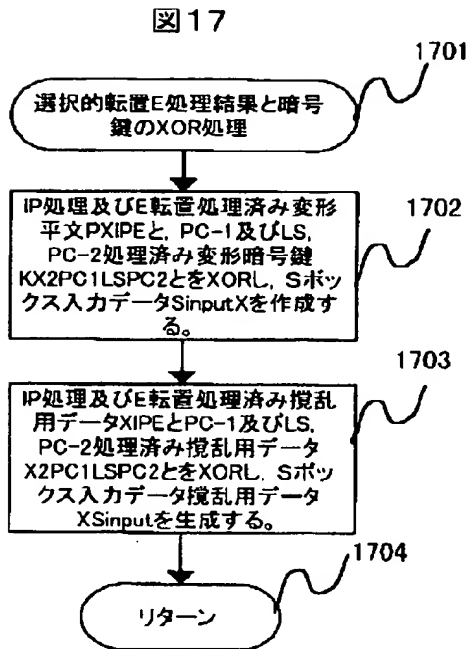


【図16】

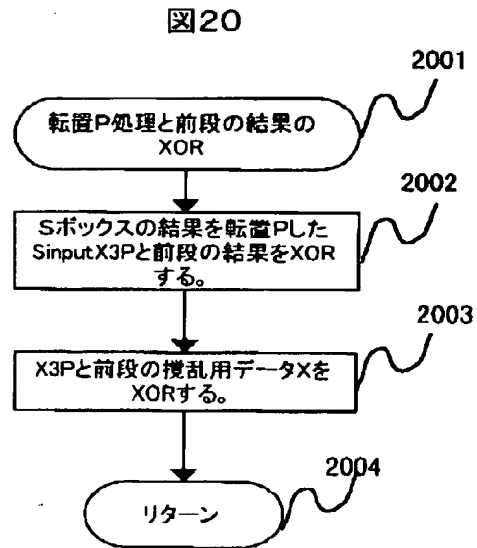
図16



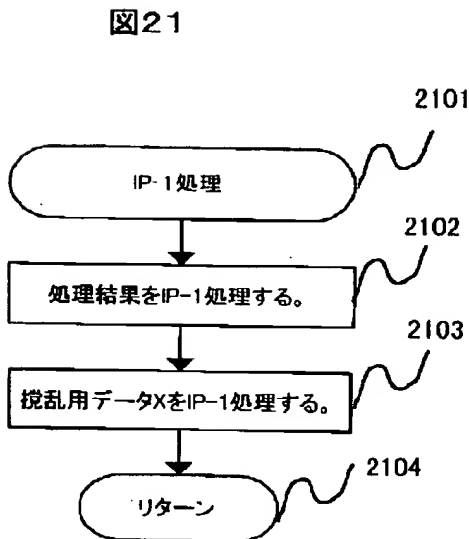
【図17】



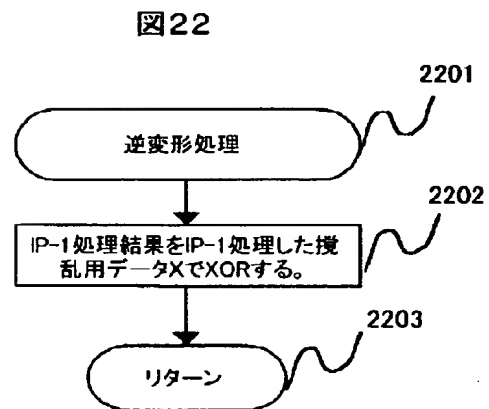
【図20】



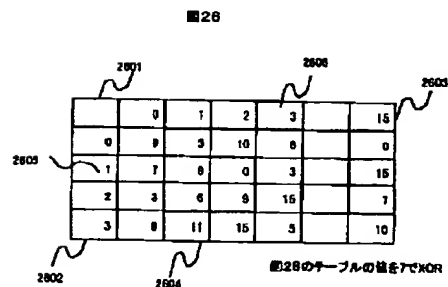
【図21】



【図22】

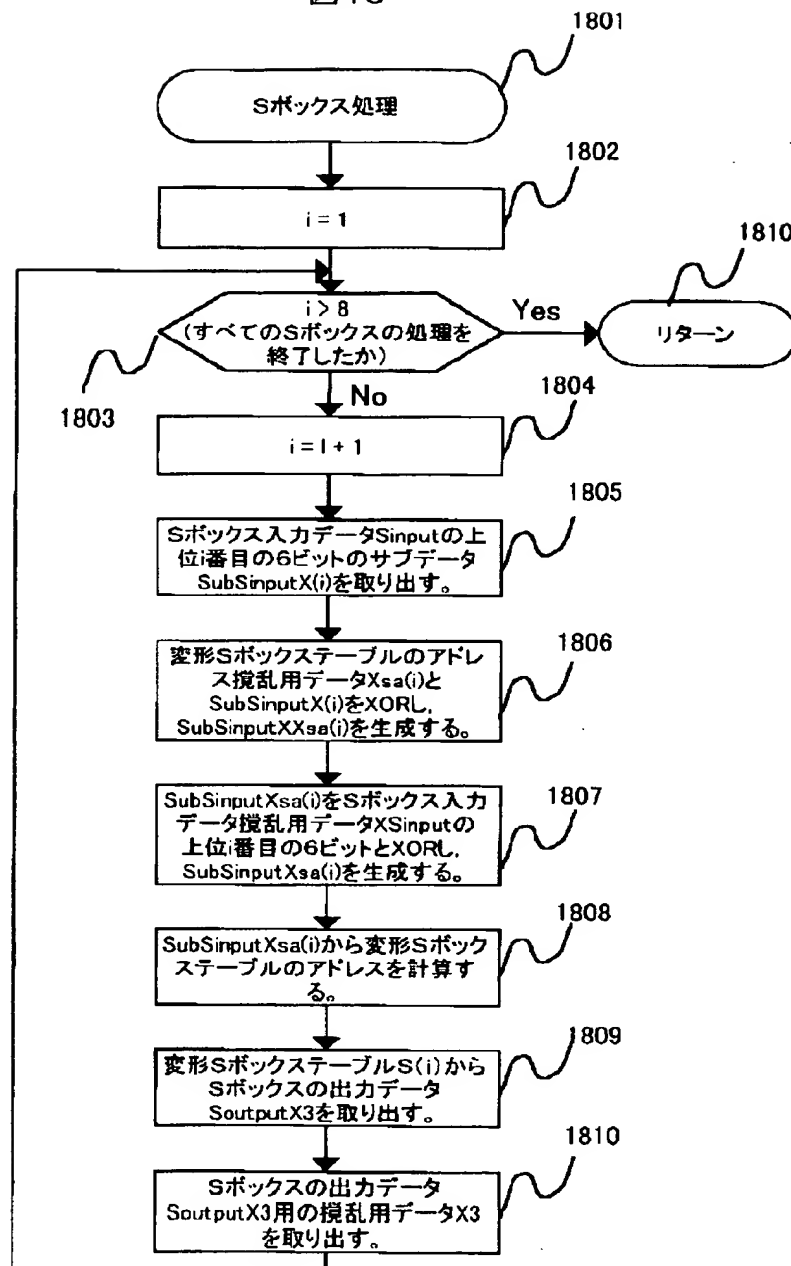


【図26】

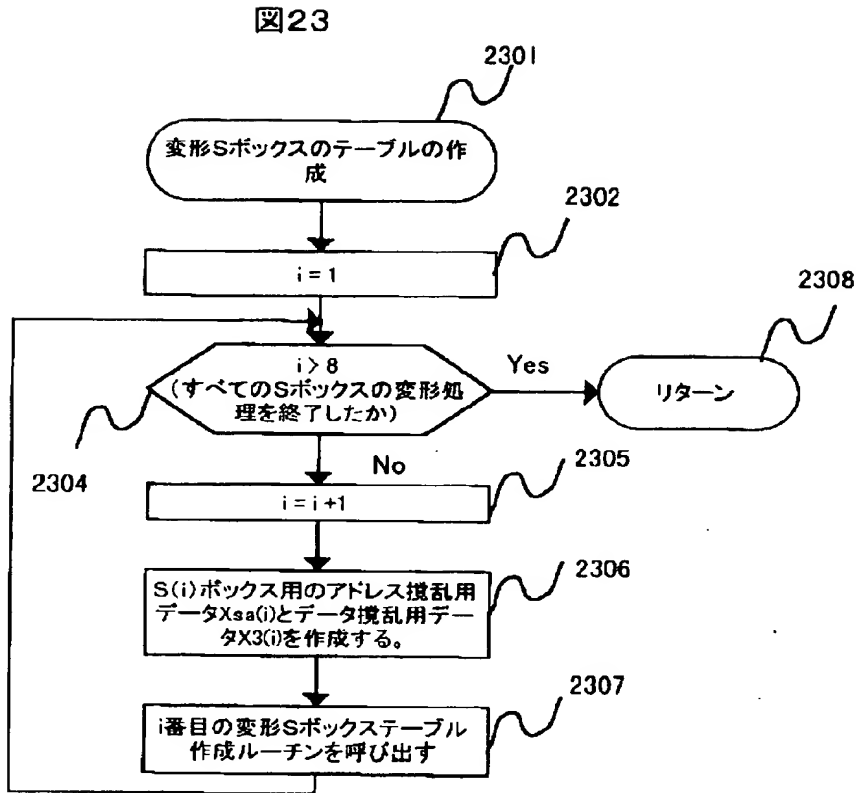


【図18】

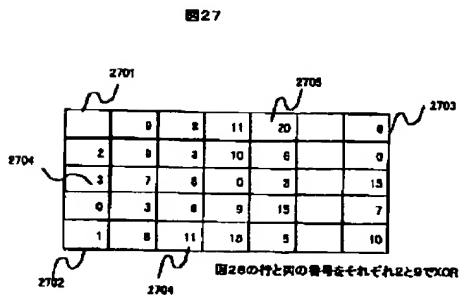
図18



【図23】



【図27】



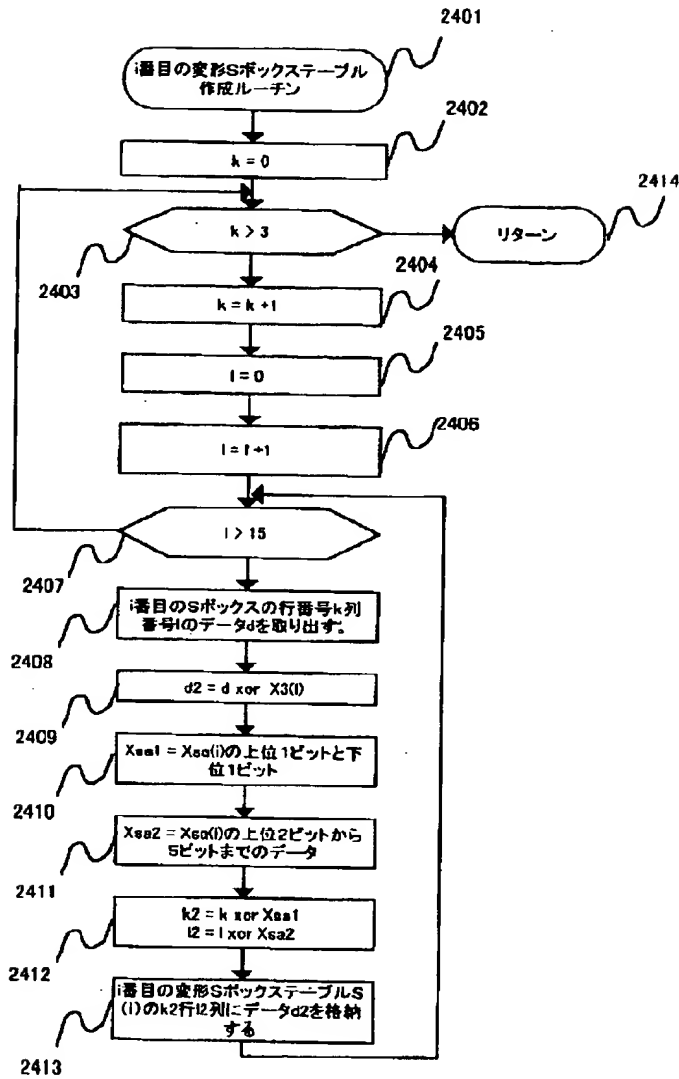
【図28】

図28

	1	2	3	4	5	6
1	32	1	2	3	4	5
2	4	5	6	7	8	9
3	6	9	10	11	12	13
4	12	13	14	15	16	17
5	16	17	18	19	20	21
6	20	21	22	23	24	25
7	24	25	26	27	28	29
8	28	29	30	31	32	1

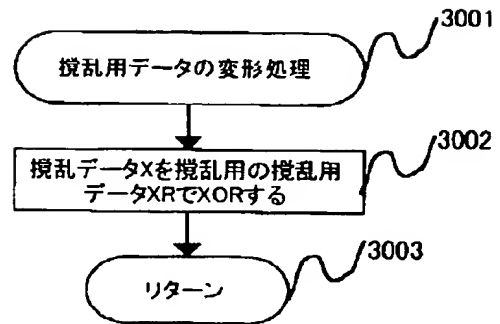
【図24】

図24



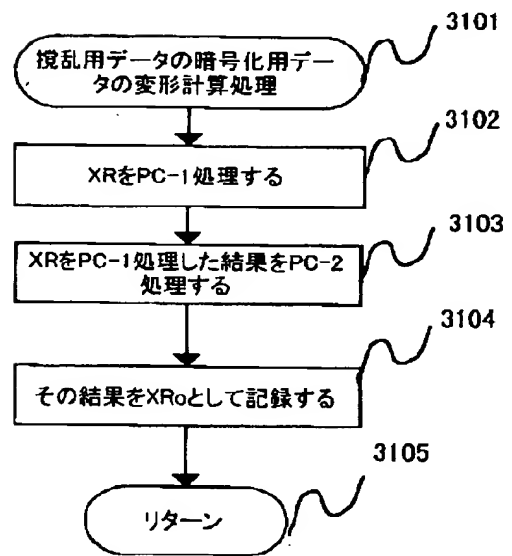
【図30】

図30



【図31】

図31



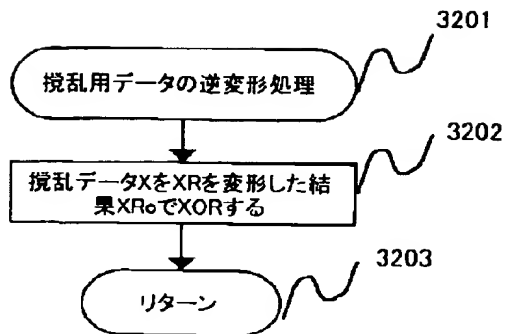
【図35】

図35

	0	1	2	3
0	5	F	7	C
1	6	B	0	3
2	D	8	4	9
3	A	2	E	1

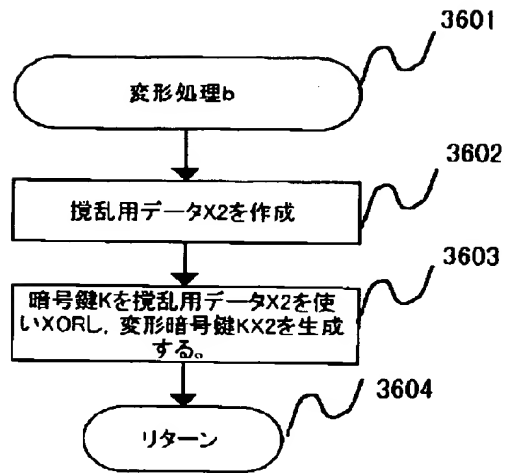
【図32】

図32



【図36】

図36



【図37】

図37

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

【図38】

図38

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

フロントページの続き

(72)発明者 奥原 進
神奈川県横浜市戸塚区戸塚町5030番地 株
式会社日立製作所ソフトウェア事業部内

(72)発明者 神永 正博
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内

45 Fターム(参考) 5B035 AA13 BB09 CA12 CA38
5J104 AA01 AA47 JA03 NA35
9A001 BB02 EE02 EE03 FF04 GG01
LL03

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING *1st page*
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.